

Problèmes inverses

Séance 4 – Déconvolution linéaire d'image

F. Orieux

orieux@l2s.centralesupelec.fr

2017 — 2018

1. Introduction

2. Problèmes inverses mal posés

En traitement de données, et en science expérimentale en particulier, on a le plus souvent des mesures indirects de nos paramètres d'intérêts. Par exemple un flux lumineux est mesuré à travers un système optique ayant sa propre réponse. Ou encore la masse d'une étoile est déduite de sa luminosité à l'aide d'un modèle physique.

Schématiquement cette relation peut s'illustrer par la figure 1. La grandeur y désigne les données effectivement mesurées. La grandeur x désigne l'ensemble des paramètres inconnus, d'intérêts ou de nuisances, pouvant être à l'origine des mesures. Le système H_θ , appelé « modèle direct », est l'élément essentiel effectuant le lien entre les inconnues et les mesures. Il peut s'agir par exemple d'un modèle instrumentale ou bien le modèle physique Λ CDM des paramètres cosmologiques vers le fond diffus cosmologique. La grandeur θ sont les paramètres influençant le comportement du modèle direct sans lien direct avec x . Au sens strict la grandeur x peut inclure θ .

Enfin, nous n'avons généralement pas de connaissance suffisante pour concevoir des modèles à même de reproduire des données réelles. La différence entre la sortie d'un modèle et les données effective est introduit et appelé dans l'immense majorité des cas « bruit ».

2.1. Situation du problème

L'objectif est d'éliminer le « flou » d'une image c'est-à-dire retrouver une image nette à partir d'une version floue de celle-ci. Ce problème entre dans la classe des problèmes d'« inversion » qui peuvent se poser lors de l'utilisation de systèmes de mesure réels. En effet, un système de mesure réel (thermomètre, caméra CCD, spectrographe ...) possède une précision finie, une dynamique finie, un temps de réponse non nul *etc.* Ainsi, la mesure fournie est toujours une version déformée (en un certain sens et dans une certaine mesure) de la grandeur physique d'intérêt.

Généralement les mesures fournies par l'instrument sont considérées suffisamment précises et fiables pour être exploitées directement. En revanche, dans d'autres situations, l'utilisation directe de ces mesures s'avère insatisfaisante. Les techniques de traitement du signal et des images peuvent alors permettre de corriger, au moins partiellement, les imperfections de ces appareils. C'est le cas qui nous intéresse ici où on peut imaginer une photo prise avec une mise au point imparfaite. L'image d'un point n'est pas un point mais une « tâche ». L'image mesurée

est alors floue puisqu'elle est la superposition des tâches issues de chaque point de l'image vraie.

Connaissant la forme de la tâche associée à un point, c'est-à-dire la réponse impulsionnelle de l'appareil et connaissant bien sûr l'image floue, le travail à réaliser consiste à retrouver l'image nette.

2.2. Mise en équation

Le modèle le plus simple pour décrire une telle transformation est un filtre linéaire et invariant, c'est-à-dire une convolution. Un tel système est représenté de façon schématique sur la figure 1.

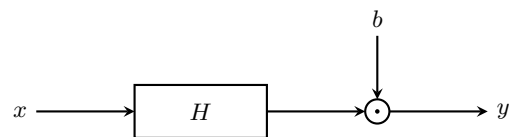


Figure 1 – Modèle direct.

Ici $x_{n,m}$ représente l'image réelle ou vraie et $y_{n,m}$ représente les données disponibles. Afin de prendre en compte les erreurs de mesures et les erreurs de modélisation, on introduit un bruit $b_{n,m}$. En l'absence d'information particulière sur la structure de ce bruit nous le supposons blanc (spatialement). L'équation d'observation correspondante prend alors la forme suivante

$$y_{n,m} = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} h_{n-p,m-q} x_{p,q} + b_{n,m} \quad (1)$$

Le travail à réaliser consiste à construire un estimateur $\hat{x}_{n,m}$ de $x_{n,m}$ à partir des mesures $y_{n,m}$.

Il est fondamental de noter qu'il s'agit ici d'un filtre passe-bas. La plupart des appareils de mesure sont d'ailleurs de ce type : ils ne peuvent pas « suivre de trop hautes fréquences ». Il est alors clair que le système élimine les composantes haute fréquence de l'image, c'est pourquoi ce problème est particulièrement difficile : il faut reconstituer les hautes fréquences alors qu'elles semblent absentes des données.

3. Déconvolution 1D

Le problème de déconvolution d'image ainsi posé est abordé dans le cadre de la théorie de l'estimation à l'aide de méthodes de moindres carrés et de régularisation quadratique. Les résultats principaux sont rappelés ci-dessous et en particulier les

critères en jeu ainsi que leur minimum qui constitue l'estimateur recherché pour l'image.

En pratique, afin d'obtenir ces estimateurs après un temps de calcul raisonnable sur un ordinateur de puissance moyenne, le sujet propose de faire quelques approximations explicitées dans la suite. Le détail de ces approximations est présenté dans la partie suivante en 1D, c'est-à-dire en déconvolution de signaux plutôt qu'en déconvolution d'image. Cette présentation permet d'étudier ces approximations et l'intérêt qu'on y trouve, de façon plus claire qu'en 2D. Le passage en 2D n'est envisagé que textuellement une extension du cas 1D. Cela dit, la partie programmation à réaliser sous IDL ne concerne que le cas 2D.

3.1. Modélisation 1D

Le modèle d'observation Eq. (1) devient en 1D

$$y_n = \sum_{p=0}^{P-1} h_p x_{n-p} + b_n.$$

On peut écrire cette relation pour N valeurs de n si on dispose de N observations et regrouper les équations obtenues sous forme de matrice

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{b}.$$

Le vecteur \mathbf{y} regroupe l'ensemble des données observées (en 2D l'image floue), le vecteur \mathbf{x} l'ensemble du signal à restaurer (en 2D l'image nette) et \mathbf{b} regroupe les échantillons du bruit. La matrice \mathbf{H} , dite matrice de convolution, possède la structure classique suivante

$$\mathbf{H} = \begin{bmatrix} h_P & \cdots & h_1 & 0 & & \\ 0 & h_P & \cdots & h_1 & 0 & \\ & 0 & \ddots & \cdots & \ddots & 0 \\ & & 0 & h_P & \cdots & h_1 \\ & & & 0 & \ddots & \vdots \\ & & & & 0 & h_P \end{bmatrix}.$$

C'est une matrice carrée, de dimension N si on dispose de N observations et qui possède une structure de Toeplitz. Le problème est donc maintenant, d'estimer le vecteur \mathbf{x} à partir du vecteur de données \mathbf{y} et de \mathbf{H} .

3.2. Moindres carrés

La technique la plus simple pour résoudre ce problème consiste à minimiser le critère de moindres carrés

$$Q_{MC}(\mathbf{x}) = (\mathbf{y} - \mathbf{H}\mathbf{x})^t (\mathbf{y} - \mathbf{H}\mathbf{x}).$$

L'estimation au sens des moindres carrés correspond au minimum d'expression

$$\hat{\mathbf{x}} = (\mathbf{H}^t \mathbf{H})^{-1} \mathbf{H}^t \mathbf{y}. \quad (2)$$

La matrice $\mathbf{H}^t \mathbf{H}$ à inverser est de taille $N \times N$ et si N est grand (ce qui est le cas en 2D pour une image) l'inversion directe est difficile pour des questions de coût de calcul. Il existe plusieurs méthodes rapides pour inverser cette matrice et calculer $\hat{\mathbf{x}}$.

3.2.1. Résolution par approximation circulante

La première méthode que l'on abordera dans ce TP est fondée sur les propriétés des matrices circulantes (consulter l'annexe). Ces propriétés permettent de « remplacer » les matrices apparaissant dans l'équation (2) par des matrices diagonales. Les produits et les inversions matriciels deviennent alors très bon marché en terme de temps de calcul. Par ailleurs, la « transformation » de ces matrices en matrices diagonales est réalisée par FFT. Cependant, pour cela il faut accepter une approximation circulante pour la matrice de convolution. Il s'agit de remplacer \mathbf{H} par

$$\tilde{\mathbf{H}} = \begin{bmatrix} h_P & \cdots & h_1 & 0 & & \\ 0 & h_P & \cdots & h_1 & 0 & \\ & 0 & \ddots & \cdots & \ddots & 0 \\ & & 0 & h_P & \cdots & h_1 \\ \ddots & & & 0 & \ddots & \vdots \\ \cdots & h_1 & & & 0 & h_P \end{bmatrix}. \quad (3)$$

Cette approximation consiste à ajouter des termes en bas et à gauche de la matrice \mathbf{H} de façon à la rendre circulante. Il faut noter que cette approximation est d'autant meilleure que N (la taille du signal ou de l'image) est plus grand que P (la longueur de la réponse impulsionnelle). En pratique, pour un signal de $N = 256$ points et une réponse impulsionnelle de $P = 10$ par exemple, cette approximation n'est pas très gênante.

La matrice $\tilde{\mathbf{H}}$ étant circulante, on peut la diagonaliser dans la base de Fourier (voir annexe)

$$\tilde{\mathbf{H}} = \mathbf{F}^\dagger \mathbf{\Lambda}_h \mathbf{F}. \quad (4)$$

La matrice $\mathbf{\Lambda}_h$ est diagonale et ses éléments diagonaux sont les valeurs propres

$$\lambda_h^1, \lambda_h^2, \dots, \lambda_h^N$$

de \mathbf{H} . Celles-ci s'obtiennent très facilement par FFT de la première ligne de \mathbf{H} c'est-à-dire par FFT de la réponse impulsionnelle calculée sur N points. Autrement dit

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \mathbf{F} \begin{bmatrix} h_1 \\ \vdots \\ h_P \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Partant de la relation Eq. (2), en exploitant la relation de diagonalisation Eq. (4) et les relations sur la matrice \mathbf{F} du cours, on obtient

$$\hat{\mathbf{x}} = (\tilde{\mathbf{H}}^t \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^t \mathbf{y} \quad (5)$$

$$\begin{aligned} &= \left(\mathbf{F}^\dagger \mathbf{\Lambda}_h^\dagger \mathbf{F} \mathbf{F}^\dagger \mathbf{\Lambda}_h \mathbf{F} \right)^{-1} \mathbf{F}^\dagger \mathbf{\Lambda}_h^\dagger \mathbf{F} \mathbf{y} \\ &= \mathbf{F}^\dagger \mathbf{\Lambda}_h^{-1} \mathbf{F} \mathbf{y}. \end{aligned} \quad (6)$$

On en déduit alors, en multipliant à gauche par \mathbf{F}

$$\mathbf{F} \hat{\mathbf{x}} = \mathbf{\Lambda}_h^{-1} \mathbf{F} \mathbf{y}.$$

On pose enfin $\hat{\mathbf{X}} = \mathbf{F} \hat{\mathbf{x}}$ et $\mathbf{Y} = \mathbf{F} \mathbf{y}$ les transformées de Fourier $\hat{\mathbf{x}}$ et \mathbf{y} . Notez que $\hat{\mathbf{X}}$ et \mathbf{Y} s'obtiennent eux aussi par FFT de $\hat{\mathbf{x}}$ et de \mathbf{y} . Le problème, dans la domaine de Fourier, devient

$$\hat{\mathbf{X}} = \mathbf{\Lambda}_h^{-1} \mathbf{Y}$$

Pour terminer, on construit le vecteur \mathbf{g}_{MC} des valeurs de la diagonale de Λ_h^{-1}

$$g_{\text{MC}}^n = 1/\lambda_n \quad \text{pour } n = 1, 2, \dots, N. \quad (7)$$

On obtient alors le vecteur $\hat{\mathbf{X}}$ par le produit « éléments par éléments » (noté \cdot et appelé produit de Hadamard) des deux vecteurs \mathbf{g}_{MC} et \mathbf{Y}

$$\hat{\mathbf{X}} = \mathbf{g}_{\text{MC}} \cdot \mathbf{Y}.$$

Remarquez que cette relation entre \mathbf{Y} (la TF de \mathbf{y}) et $\hat{\mathbf{X}}$ (la TF de \mathbf{x}) est une relation de « filtrage dans le domaine de Fourier » dont la fonction de « transfert en fréquence discrète » est \mathbf{g}_{MC} . En résumé la procédure à suivre est donnée est

1. Construire la FFT λ_h de la réponse impulsionnelle sur N points.
2. Construire le vecteur \mathbf{g}_{MC} du transfert en fréquence selon l'équation (7).
3. Construire la FFT \mathbf{Y} des données.
4. Calculer $\hat{\mathbf{X}}$ produit du transfert en fréquence \mathbf{g}_{MC} par \mathbf{Y} .
5. Revenir dans le domaine spatial par FFT inverse de $\hat{\mathbf{X}}$ pour obtenir $\hat{\mathbf{x}}$.

3.2.2. Algorithme de descente selon le gradient

La deuxième méthode que l'on abordera dans ce TP est une méthode de type descente de gradient. Comme le critère est quadratique, il est convexe et ne possède qu'un minimum global cf. Eq. (2). Pour trouver le minimum du critère, quelque soit le point de départ on cherche la direction de plus forte pente (c'est ce qu'on appelle la direction du gradient), puis on se déplace dans cette direction. A chaque répétition on se rapproche du minimum.

Rappelons tout d'abord l'expression du critère Q_{MC}

$$\begin{aligned} Q_{\text{MC}}(\mathbf{x}) &= (\mathbf{y} - \mathbf{H}\mathbf{x})^t (\mathbf{y} - \mathbf{H}\mathbf{x}) \\ &= \mathbf{x}^t \mathbf{H}^t \mathbf{H} \mathbf{x} - 2\mathbf{x}^t \mathbf{H}^t \mathbf{y} + \mathbf{y}^t \mathbf{y}. \end{aligned}$$

Pour calculer le gradient du critère on le dérive par rapport au vecteur \mathbf{x}

$$\begin{aligned} \frac{\partial Q_{\text{MC}}(\mathbf{x})}{\partial \mathbf{x}} &= 2\mathbf{H}^t \mathbf{H} \mathbf{x} - 2\mathbf{H}^t \mathbf{y} \\ &= 2\mathbf{H}^t (\mathbf{H}\mathbf{x} - \mathbf{y}). \end{aligned}$$

Notons par $\mathbf{e}_k = (\mathbf{H}\mathbf{x} - \mathbf{y})$, l'erreur entre la sortie modèle et les données. Pour mettre en œuvre l'algorithme on prend un \mathbf{x}^0 quelconque puis on applique l'itération suivante

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \mathbf{H}^t (\mathbf{H}\mathbf{x}^k - \mathbf{y}). \quad (8)$$

Si

$$\alpha < \frac{2}{\|\mathbf{H}^t \mathbf{H}\|_M},$$

alors l'algorithme converge vers le minimum du critère où $\|\cdot\|_M$ est la norme d'une matrice égale au module au carré de la plus grande des valeurs singulières. En résumé la procédure à suivre est

1. Initialiser \mathbf{x}^0 à une valeur quelconque.
2. Calculer $\mathbf{H}\mathbf{x}^k$ en convoluant \mathbf{x}^k par la réponse impulsionnelle.
3. Calculer l'erreur \mathbf{e}_k , $(\mathbf{H}\mathbf{x} - \mathbf{y})$.
4. Calculer $\mathbf{H}^t (\mathbf{H}\mathbf{x}^k - \mathbf{y})$ en convoluant le résultat de l'étape précédente par la réponse impulsionnelle retournée.
5. Multiplier le résultat de l'étape précédente par α et le soustraire à \mathbf{x}^k .
6. Retourner à l'étape 2 tant que l'algorithme n'a pas convergé.

3.3. Régularisation ou pénalisation

Quand vous aurez utilisé la méthode des moindres carrés vous constaterez qu'elle n'est pas toujours satisfaisante. Dans certains cas l'image obtenue reste très bruitée, très chahutée. Il vous appartient d'expliquer pourquoi et c'est d'ailleurs une des questions qui vous est posée à la section 4.2.1.

Quoiqu'il en soit, pour corriger ce défaut il est possible de modifier le critère des moindres carrés afin d'assurer une certaine régularité à l'image obtenue. L'idée la plus simple consiste à pénaliser les fortes différences entre échantillons successifs du signal à restaurer. Le critère prend alors la forme régularisée suivante

$$\begin{aligned} Q_{\text{MCR}}(\mathbf{x}) &= (\mathbf{y} - \mathbf{H}\mathbf{x})^t (\mathbf{y} - \mathbf{H}\mathbf{x}) + \mu \sum_{n=1}^{N-1} (x_{n+1} - x_n)^2 \\ &= (\mathbf{y} - \mathbf{H}\mathbf{x})^t (\mathbf{y} - \mathbf{H}\mathbf{x}) + \mu \mathbf{x}^t \mathbf{D}^t \mathbf{D} \mathbf{x} \end{aligned}$$

avec \mathbf{D} la matrice de différences d'ordre 1, de taille $(N-1) \times N$, définie comme

$$\mathbf{D} = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & \dots & 0 \\ \vdots & & \ddots & \ddots & & & \vdots \\ \vdots & & & \ddots & \ddots & & \vdots \\ 0 & \dots & \dots & 0 & 1 & -1 & 0 \\ 0 & \dots & \dots & 0 & 0 & 1 & -1 \end{bmatrix}.$$

Ce critère est, comme celui des moindres carrés simples, quadratique en \mathbf{x} et on connaît l'expression de son minimum

$$\hat{\mathbf{x}} = (\mathbf{H}^t \mathbf{H} + \mu \mathbf{D}^t \mathbf{D})^{-1} \mathbf{H}^t \mathbf{y}.$$

3.3.1. Approximation circulante

Pour les mêmes raisons que dans le cas des moindres carrés simples, on accepte une approximation circulante pour \mathbf{H} et \mathbf{D} de manière à « remplacer » les calculs sur des matrices quelconques par des calculs sur des matrices diagonales. En ce qui concerne \mathbf{H} on conserve l'approximation par $\tilde{\mathbf{H}}$ donnée par Eq. (3). En ce qui concerne \mathbf{D} il suffit d'y ajouter une dernière ligne pour obtenir une approximation circulante $\tilde{\mathbf{D}}$, de dimension $N \times N$

$$\tilde{\mathbf{D}} = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & \dots & 0 \\ \vdots & & \ddots & \ddots & & & \vdots \\ \vdots & & & \ddots & \ddots & & \vdots \\ 0 & \dots & \dots & 0 & 1 & -1 & 0 \\ 0 & \dots & \dots & 0 & 0 & 1 & -1 \\ -1 & \dots & \dots & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Le minimum approché prend alors la forme suivante

$$\hat{\mathbf{x}} = (\tilde{\mathbf{H}}^t \tilde{\mathbf{H}} + \mu \tilde{\mathbf{D}}^t \tilde{\mathbf{D}})^{-1} \tilde{\mathbf{H}}^t \mathbf{y}$$

En partant de cette expression, en exploitant les relations sur la matrice \mathbf{F} contenus en annexe et en vous inspirant des calculs du cas des moindres carrés simples des relations 5 à 6, montrez que

$$\hat{\mathbf{x}} = \mathbf{F}^\dagger (\Lambda_h^\dagger \Lambda_h + \Lambda_d^\dagger \Lambda_d)^{-1} \Lambda_h^\dagger \mathbf{F} \mathbf{y}$$

avec Λ_h et Λ_d les matrices diagonales des valeurs propres de $\tilde{\mathbf{H}}$ et $\tilde{\mathbf{D}}$. On continue à noter $\lambda_h = [\lambda_h^1, \lambda_h^2, \dots, \lambda_h^N]$ le vecteur des valeurs propres de $\tilde{\mathbf{H}}$ et on note $\lambda_d = [\lambda_d^1, \lambda_d^2, \dots, \lambda_d^N]$ le

vecteur des valeurs propres de \mathbf{D} . Indiquez dans votre compte rendu comment on obtient ces dernières.

Comme précédemment, posez $\hat{\mathbf{X}} = \mathbf{F}\hat{\mathbf{x}}$ et $\mathbf{Y} = \mathbf{F}\mathbf{y}$ les transformées de Fourier $\hat{\mathbf{x}}$ et \mathbf{y} et déduisez

$$\hat{\mathbf{X}} = (\Lambda_h^\dagger \Lambda_h + \mu \Lambda_d^\dagger \Lambda_d)^{-1} \Lambda_h^\dagger \mathbf{Y}.$$

Pour terminer, construisez le vecteur \mathbf{g}_{MCR} des valeurs de la diagonale de la matrice $(\Lambda_h^\dagger \Lambda_h + \mu \Lambda_d^\dagger \Lambda_d)^{-1} \Lambda_h^\dagger$

$$\mathbf{g}_{\text{MCR}}^n = \frac{(\lambda_h^n)^*}{|\lambda_h^n|^2 + \mu |\lambda_d^n|^2} \quad \text{pour } n = 1, 2, \dots, N. \quad (9)$$

Expliquez alors pourquoi le vecteur $\hat{\mathbf{X}}$ s'obtient comme le produit « éléments par éléments » des deux vecteurs \mathbf{g}_{MCR} et \mathbf{Y}

$$\hat{\mathbf{X}} = \mathbf{g}_{\text{MCR}} \cdot \mathbf{Y}$$

Remarquez qu'il s'agit ici encore d'une relation de « filtrage dans le domaine de Fourier » dont la fonction de « transfert discrète » est \mathbf{g}_{MCR} . En résumé la procédure à suivre est

1. Construire la FFT λ_h de la réponse impulsionnelle sur N points.
2. Construire la FFT λ_d de $[1, -1]$ sur N points.
3. Construire le vecteur \mathbf{g}_{MCR} du transfert en fréquence selon l'Eq. (9).
4. Construire la FFT \mathbf{Y} des données.
5. Calculer $\hat{\mathbf{X}}$ comme produit du transfert en fréquence \mathbf{g}_{MCR} par \mathbf{Y} .
6. Revenir dans le domaine spatial par FFT inverse de $\hat{\mathbf{X}}$ pour obtenir $\hat{\mathbf{x}}$.

3.3.2. Algorithme de descente selon le gradient

Réécrivons les équations de la partie 3.2.2 avec le nouveau critère Q_{MCR}

$$\begin{aligned} Q_{\text{MCR}}(\mathbf{x}) &= (\mathbf{y} - \mathbf{H}\mathbf{x})^t (\mathbf{y} - \mathbf{H}\mathbf{x}) + \mu \mathbf{x}^t \mathbf{D}^t \mathbf{D} \mathbf{x} \\ &= \mathbf{x}^t \mathbf{H}^t \mathbf{H} \mathbf{x} - 2\mathbf{x}^t \mathbf{H}^t \mathbf{y} + \mathbf{y}^t \mathbf{y} + \mu \mathbf{x}^t \mathbf{D}^t \mathbf{D} \mathbf{x}. \end{aligned}$$

Pour calculer le gradient du critère on le dérive par rapport au vecteur \mathbf{x}

$$\frac{\partial Q_{\text{MCR}}(\mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{H}^t \mathbf{H} \mathbf{x} - 2\mathbf{H}^t \mathbf{y} + 2\mu \mathbf{D}^t \mathbf{D} \mathbf{x} \quad (10)$$

$$= 2\mathbf{H}^t (\mathbf{H} \mathbf{x} - \mathbf{y}) + 2\mu \mathbf{D}^t \mathbf{D} \mathbf{x}. \quad (11)$$

Pour la mise en œuvre de l'algorithme on prend un \mathbf{x}^0 quelconque puis on applique l'itération suivante

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \left(\mathbf{H}^t (\mathbf{H} \mathbf{x}^k - \mathbf{y}) + \mu \mathbf{D}^t \mathbf{D} \mathbf{x}^k \right) \quad (12)$$

où le pas de descente optimal est

$$\alpha_{\text{opt}} = \frac{\mathbf{g}_k^t \mathbf{g}_k}{\mathbf{g}_k^t (\mathbf{H}^t \mathbf{H} + \mu \mathbf{D}^t \mathbf{D}) \mathbf{g}_k} \quad (13)$$

avec \mathbf{g}_k est le gradient en \mathbf{x}_k

$$\mathbf{g}_k = \left(\mathbf{H}^t (\mathbf{H} \mathbf{x}^k - \mathbf{y}) + \mu \mathbf{D}^t \mathbf{D} \mathbf{x}^k \right). \quad (14)$$

En résumé la procédure à suivre est

1. Initialiser \mathbf{x}^0 à une valeur quelconque.
2. Calculer $\mathbf{H} \mathbf{x}^k$ en convoluant \mathbf{x}^k par la réponse impulsionnelle.

3. Calculer la différence \mathbf{e}_k entre $\mathbf{H} \mathbf{x}^k$ et les données \mathbf{y} .
4. Calculer $\mathbf{H}^t (\mathbf{H} \mathbf{x}^k - \mathbf{y})$ en convoluant le résultat de l'étape précédente par la réponse impulsionnelle retournée.
5. Calculer $\mathbf{D} \mathbf{x}^k$ en convoluant \mathbf{x}^k par le filtre dérivateur $[-1, 1]$.
6. Calculer $\mathbf{D}^t \mathbf{D} \mathbf{x}^k$ en convoluant $\mathbf{D} \mathbf{x}^k$ par le filtre dérivateur retournée $[1, -1]$.
7. Mettre à jour \mathbf{x}^k en utilisant l'Eq. (12).
8. Retourner à l'étape 2 tant que l'algorithme n'a pas convergé,

3.4. Passage en 2D

3.4.1. Approches circulantes

On peut développer des calculs rigoureux analogues au cas 1D pour établir les résultats du cas 2D. Cependant, ces calculs sont beaucoup plus fastidieux car les matrices ont une structure plus complexe structure de Toëplitz par bloc, chaque bloc possédant une structure de Toëplitz. Par ailleurs, l'approximation circulante dans les deux directions est plus difficile à manipuler. C'est pourquoi, nous n'abordons le cas 2D comme une extension du cas 1D. Quelques remarques cependant.

- Les images, la réponse impulsionnelle, les termes de douceurs sont maintenant bidimensionnels, c'est pourquoi les FFT sont remplacées par des FFT2D.
- Plus précisément, si l'image à traiter possède N lignes et N colonnes toutes les FFT2D doivent être prises sur N lignes et N colonnes.
- Le transfert en fréquence est lui aussi bidimensionnel : une dimension pour chacun des deux fréquences spatiales.
- L'opérateur \mathbf{D} se divise en 2D en deux opérateurs \mathbf{D}_1 et \mathbf{D}_2 qui fond une dérivée 1D respectivement sur les lignes et sur les colonnes de l'image.

Les structures des procédures déduite du cas 1D sont indiquées ci-dessous, d'abord dans le cas des moindres carrés simples puis dans le cas des moindres carrés régularisés.

1. Construire λ_h la FFT2D de la réponse impulsionnelle sur $N \times N$ points.
2. Construire la matrice \mathbf{g}_{MC} du transfert en fréquence selon 7.
3. Construire \mathbf{Y} la FFT2D des données.
4. Calculer $\hat{\mathbf{X}}$ texttt produit \mathbf{g}_{MC} par \mathbf{Y} .
5. Revenir dans le domaine spatial par FFT2D inverse de $\hat{\mathbf{X}}$ pour obtenir $\hat{\mathbf{x}}$.

1. Construire λ_h la FFT2D de la réponse impulsionnelle sur $N \times N$ points.
2. Construire λ_{d1} la FFT2D de $(0 \ 1 \ -1)$ sur $N \times N$ points.
3. Construire λ_{d2} la FFT2D de

$$\begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$$

sur $N \times N$ points.

4. Construire la matrice \mathbf{g}_{MCR} du transfert en fréquence selon 9.
5. Construire \mathbf{Y} la FFT2D des données.
6. Calculer $\hat{\mathbf{X}}$ texttt produit de \mathbf{g}_{MC} par \mathbf{Y} .

7. Revenir dans le domaine spatial par FFT2D inverse de \hat{X} pour obtenir \hat{x} .

L'extension 2D de la méthode de descente de gradient se fait simplement, en remplaçant des opération de convolution par des convolutions 2D à l'aide de la commande `convolution` et en utilisant la convolution avec

$$D_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{et} \quad D_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}.$$

4. Déconvolution d'image

4.1. Image à traiter

Charger sous IDL les données fournies dans les fichiers `donnee1.sav` pour l'algorithme sous approximation circulante et `donnee2.sav` pour l'algorithme de descente suivant le gradient. Ces fichiers contiennent deux images, l'image flou (nos données `data`) l'image parfaite `photo` et la réponse impulsionnelle du filtre `fftker` dans l'approche circulante et `ker` dans l'approche descente selon le gradient. Les images contenues dans `donnee1.sav` et `donnee2.sav` correspondent à une photo d'un photographe on se servira de ces données pour valider nos algorithmes et étudier les problèmes de bord.

Tracez les réponses impulsionnelles $h_{n,m}$ et les transferts en fréquence $H(\nu_x, \nu_y)$ correspondant. Prenez le temps d'identifier les deux axes fréquentiels, les fréquences nulles, les fréquences maximales, etc. Faites apparaître clairement ces résultats dans le compte rendu. *Quelle est la nature du filtre ?*

4.2. Mise en œuvre de la déconvolution

Les programmes IDL à réaliser doivent mettre en œuvre la déconvolution à l'aide de la minimisation des deux critères : les moindres carrés simples et les moindres carrés régularisés, en 2D.

4.2.1. Inversion par approximation circulante

Vous en arrivez à la partie la plus importante de cette séance : analyse et commentaires des résultats.

Notez, bien sûr, l'aspect général de l'image obtenue et observez en particulier la netteté des frontières par exemple entre le manteau du photographe et le ciel. Intéressez-vous aussi aux détails du visage ou de l'appareil photo.

Que se passe-t-il avec le filtre inverse et pourquoi ? Avec la méthode régularisée, expliquez qualitativement les résultats obtenus. Quelle est l'influence du paramètre μ ? Confrontez les résultats obtenus aux résultats attendus compte tenu de la forme du terme de régularisation.

4.2.2. Inversion à l'aide d'un algorithme de descente de gradient

Testez votre algorithme sur le critère Q_{MC} avec différentes valeurs de α : $\alpha = 1,5$, $\alpha = 0,015$, $\alpha = 2,03$. Que remarquer vous ? Testez votre algorithme sur le critère Q_{MCR} avec $\mu = 0,005$ et $0 < \alpha < 0,2$. Comparez le résultat obtenu, avec le résultat précédent, en particulier sur les bords de l'image. Comparez aussi le temps de calcul des deux approches. Codez la fonction qui calcule l'alpha optimal à chaque étape.

A. Diagonalisation des matrices circulantes

Cette annexe réunit les résultats d'algèbre linéaire concernant la diagonalisation des matrices circulantes dans la base de Fourier. Ces résultats permettent la mise en œuvre rapide de la méthode proposée.

A.1. Matrices circulantes

Décalage circulaire (ou rotation) d'un vecteur

Pour effectuer le décalage circulaire (ou rotation) d'un vecteur il suffit de décaler ses éléments d'une place vers la droite et de replacer le dernier élément en première position, de la façon suivante

$$[c_1, c_2, c_3, \dots, c_{N-1}, c_N] \longrightarrow [c_N, c_1, c_2, c_3, \dots, c_{N-1}].$$

Matrice circulante

On dit qu'une matrice est circulante si on obtient chacune de ses lignes par décalage circulaire de la ligne précédente, la première ligne s'obtenant par décalage circulaire de la dernière. Une matrice circulante est donc du type

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_N \\ c_N & c_1 & c_2 & & c_{N-1} \\ c_{N-1} & c_N & c_1 & \ddots & c_{N-2} \\ \vdots & & & \ddots & \vdots \\ c_3 & c_4 & c_5 & & c_2 \\ c_2 & c_3 & c_4 & \dots & c_N & c_1 \end{bmatrix}.$$

En premier lieu, on constate que cette matrice possède une structure de Toeplitz, c'est-à-dire que ses éléments sont constants sur les diagonales : c_1 sur la diagonale, c_2 sur la première diagonale supérieure, etc. et inversement c_N sur la première diagonale inférieure, etc. On constate également que chaque colonne s'obtient par décalage circulaire de la colonne précédente, la première colonne s'obtenant par décalage circulaire de la dernière. On peut donc construire une matrice circulante en décalant les lignes ou en décalant les colonnes.

La mise en œuvre rapide des méthodes proposées dans ce TP exploite le résultat suivant : les matrices circulantes sont diagonalisables dans la base de Fourier. Ce résultat permettra de remplacer des produits matriciels quelconques par des produits de matrices diagonales (beaucoup plus économique), au prix de quelques calculs de FFT.

A.2. Matrice de Fourier et quelques propriétés

Notons F la matrice de Fourier, carrée de taille N , c'est-à-dire la « matrice de FFT » sur N points d'un vecteur de taille N . Elle est caractérisée par : $X = Fx$ où

- x est un vecteur quelconque de taille N et
- X le vecteur de taille N , des N échantillons de sa transformée de Fourier sur la grille fréquentielle régulière de N points entre 0 et $1 - 1/N$.

Cette matrice possède des propriétés particulières, utiles pour les calculs de la section 3.1

$$\begin{aligned} F^t &= F \\ FF^\dagger &= I_N \\ F^{-1} &= F^\dagger = F^*. \end{aligned}$$

A.3. Diagonalisation

On montre que \mathbf{C} est diagonalisable dans la base de Fourier \mathbf{F} , c'est-à-dire

$$\mathbf{C} = \mathbf{F}^\dagger \mathbf{\Lambda}_C \mathbf{F}$$

où $\mathbf{\Lambda}_C$ est la matrice diagonale des valeurs propres de \mathbf{C} . On montre également que ces valeurs propres s'obtiennent comme la TF de la première ligne $\mathbf{c} = [c_1, c_2, \dots, c_N]^t$ de la matrice \mathbf{C}

$$\lambda_C = \mathbf{F} \mathbf{c}$$

où λ_C est le vecteur contenant les N valeurs propres de \mathbf{C} , c'est-à-dire les éléments diagonaux de $\mathbf{\Lambda}_C$. Le vecteur λ_C s'obtient évidemment par FFT du vecteur \mathbf{c} .