Regularization of tensorial inverse problems via convex optimization : Applications in image and video processing

Oumaima Benchettou

Thèse de Doctorat

Mention : Mathématiques et leurs interactions
Spécialité : Mathématiques Appliquées

présentée à ***l'Ecole Doctorale Sciences, Technologie, Santé (ED n°585)***
de l'Université du Littoral Côte d'Opale
et
au Centre d'études doctorales Sciences de l'Ingénieur
de l'Université Cadi Ayyad Marrakech

par

Benchettou Oumaima

pour obtenir le grade de Docteur de l'Université du Littoral Côte d'Opale

Régularisation des problèmes inverses tensoriels par l'optimisation convexe:

Applications au traitement des images et des vidéos

Soutenue le 27 décembre 2022, après avis des rapporteurs,

devant le jury d'examen :

Hassane Sadok,	**Professeur, Université du Littoral Côte d'Opale**	**Président**
Abdellah Bnouhachem,	**Professeur, Université Ibn Zohr, Agadir**	**Rapporteur**
Hassan Safouhi,	**Professeur, Université d'Alberta, Canada**	**Rapporteur**
Otmane Souhar,	**PH, Université Chouaib Doukkali, El Jadida**	**Rapporteur**
Abderrahman Bouhamidi,	**Professeur, Université du Littoral Côte d'Opale**	**Directeur**
Abdeslem Hafid Bentbib,	**Professeur, Université Cadi Ayyad, Marrakech**	**Directeur**

Thèse de Doctorat

Mention : Mathématiques et leurs interactions
Spécialité : Mathématiques Appliquées

présentée à ***l'Ecole Doctorale Sciences, Technologie, Santé (ED n°585)***
de l'Université du Littoral Côte d'Opale
et
au Centre d'études doctorales Sciences de l'Ingénieur
de l'Université Cadi Ayyad Marrakech

par

Benchettou Oumaima

pour obtenir le grade de Docteur de l'Université du Littoral Côte d'Opale

Regularization of tensorial inverse problems via convex optimization:

Applications in image and video processing.

Soutenue le 27 décembre 2022, après avis des rapporteurs,

devant le jury d'examen :

Hassane Sadok,	**Professeur, Université du Littoral Côte d'Opale**	**Président**
Abdellah Bnouhachem,	**Professeur, Université Ibn Zohr, Agadir**	**Rapporteur**
Hassan Safouhi,	**Professeur, Université d'Alberta, Canada**	**Rapporteur**
Otmane Souhar,	**PH, Université Chouaib Doukkali, El Jadida**	**Rapporteur**
Abderrahman Bouhamidi,	**Professeur, Université du Littoral Côte d'Opale**	**Directeur**
Abdeslem Hafid Bentbib,	**Professeur, Université Cadi Ayyad, Marrakech**	**Directeur**

This thesis has been prepared at the following research units.

Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville (LMPA)
50, rue Ferdinand Buisson
CS 80699
62228 Calais Cedex
France

☎ +333 21 46 55 89
✉ secretariat@lmpa.univ-littoral.fr
Web Site https://lmpa.univ-littoral.fr/

Laboratoire de Mathématiques Appliquées et Informatique (LAMAI)
UCAM, FSTG,
BP 549 Marrakech
Maroc

☎ +212 5 24 43 34 04
🖷 +212 5 24 43 31 70
Web Site http://www.fstg-marrakech.ac.ma/lamai/

Regularization of tensorial inverse problems via convex optimization
Applications in image and video processing

Abstract

Solving inverse problems in a multidimensional setting has become an active topic on which many researchers in linear algebra are working. On the one hand, the construction of a higher dimensional model can be achieved using tensor algebra by adopting the mechanisms developed recently in this field. On the other hand, the solution to such problems is usually based on the regularization techniques that remedy the ill-conditioning that can be exhibited in almost all inverse problems.
The present thesis aims to bring together the modeling of inverse problems in a higher dimension and the generalization of some variational regularization methods in tensorial form. Recently, the variational regularization methods are known as well-established methods for solving inverse problems. For example, Tikhonov and total variation regularizers are among the well-known approaches that we will generalize and develop in the tensor form. Convex optimization approaches will play an essential role in the resolution of the constrained regularization problems that we have proposed. As well as a set of mechanisms, such as projection methods and extrapolation techniques, which have contributed to enhancing the performance of the developed algorithms. Numerical applications in image and video processing are given to illustrate the effectiveness of the proposed approaches compared with some state-of-art methods.

Keywords: tensorial algebra, convex optimization, regularization methods, Tickonov, total variation, projection methods, extrapolation techniques, image and video processing.

Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville (LMPA) – 50, rue Ferdinand Buisson – CS 80699 – 62228 Calais Cedex – France
Laboratoire de Mathématiques Appliquées et Informatique (LAMAI) – UCAM, FSTG, – BP 549 Marrakech – Maroc

Régularisation des problèmes inverses tensoriels par l'optimisation convexe
Applications au traitement des images et des vidéos

Résumé

La résolution des problèmes inverses dans un cadre multidimensionnel est devenue un sujet actif sur lequel travaillent de nombreux chercheurs en algèbre linéaire. D'une part, la construction d'un modèle en dimension supérieure peut être réalisée en utilisant l'algèbre tensorielle en adoptant les mécanismes développés récemment dans ce domaine. D'autre part, la résolution de tels problèmes est généralement basée sur l'utilisation de techniques de régularisation qui remédient au mauvais conditionnement que l'on peut trouver dans presque tous les problèmes inverses.
La présente thèse vise à réunir la modélisation des problèmes inverses en dimension supérieure et la généralisation de certaines méthodes de régularisation variationnelle sous forme tensorielle. Les méthodes de régularisation variationnelle sont connues comme des méthodes bien établies pour résoudre les problèmes inverses. Par exemple, les régularisateurs de Tikhonov et de la variation totale font partie des approches bien connues que nous généraliserons et développerons sous forme tensorielle. Les approches d'optimisation convexe joueront un rôle essentiel dans la résolution des problèmes de régularisation sous contraintes que nous avons proposés. Ainsi qu'un ensemble de mécanismes, tels que les méthodes de projection et les techniques d'extrapolation, qui ont contribué à améliorer les performances des approches développées. Des applications numériques dans le traitement des images et des vidéos sont données pour illustrer l'efficacité des approches proposées par rapport à certaines méthodes de l'état de l'art.

Mots clés : algèbre tensorielle, optimisation convexe, méthodes de régularisation, Tickonov, variation totale, méthodes de projection, techniques d'extrapolation, traitement des images et des vidéos.

To my parents and family!

To my dearly directors!

To friends and colleagues!

I also dedicate this work
to all those who took part in this journey.

Acknowledgments

It was an exceptional journey during which I learned a lot and I am still learning, both on a personal and academic level. The credit is due to the people who participated and supported me on this path.

First and foremost I am extremely grateful to my supervisors, Professor Abdeslem Hafid Bentbib and Professor Abderrahman Bouhamidi for their invaluable advice, continuous support, and patience during my PhD study. Without a doubt, this thesis would be impossible without them. Their immense knowledge and plentiful experience have encouraged me all the time in my academic research and daily life. Their continued support over the years allowed me to develop my writing and research skills and encouraged me to open up to international forums that promote scientific discussion and exchanges.

I would like to thank the reviewers of this thesis Professors, Bnouhachem Abdellah, ENSA Université Ibn Zohr, Safouhi Hassan, Université Alberta, Souhar Otmane, Université Chouaib Doukkali, El Jadida for the interest they have shown in my work. Thanks should also go to Professor Hassane Sadok for having accepted to be part of my thesis jury as well as for supporting me during my academic experience at the University Littoral Côte d'Opale. Many thanks to the entire EMA team for welcoming me to the LMPA-UCLO Calais research unit and allowing me to work in good conditions. I would be remiss in not mentioning the LAMIA-UCA research unit which allowed me, thanks to an allocation of research and various cognitive, physical and moral aids, to devote myself serenely to the development of my thesis.

Special thanks to Professor Kriet Karim, Professor at Cadi Ayyad University, for his cooperation with me in my work, for his feedbacks and encouragement, as well as for his sympathy and kindness. Special gratitude to my friend, colleague, and roommate, Khouia Asmaa, for sharing this eight-year academic journey with

me. I would like to thank all my friends and colleagues in the two labs for their encouragement and enthusiasm.

I would like to express my gratitude to the most precious people in my world, my parents Aicha and El Mustapha, for loving, helping and supporting me from the very first step until now, without them I couldn't have done it, also my beloved sisters Soukaina and Salma for their unlimited love. Thank you to all the members of my family, without them I could never complete my achievements.

Table of contents

List of Tables

List of Figures

List of Acronyms

List of notations and symbols

$\mathbb{R}$	Field of real numbers
$\mathbb{R}^{m\times n}$	Space of real $m \times n$ matrices
$\mathbb{R}^{I_1\times\ldots\times I_N}$	Space of real $I_1 \times \ldots \times I_N$ tensor.
ℓ^1	The space of sequences whose series is absolutely convergent.
$L^1(\Omega)$	Space of integrable functions over the set Ω.
$L^2(\Omega)$	The Hilbert space of square summable function over the set Ω.
$L^p(\Omega)$	Lebesgue spaces over the set Ω.
$\mathcal{C}_c^1(\Omega, \mathbb{R}^n)$	Space of continuously differentiable functions with compact support in Ω.
$W^{m,p}(\Omega)$	The Sobolev space over the set Ω.
$BV(\Omega)$	Space of functions of bounded variation.
$\mathcal{L}(E, F)$	The set of linear maps from E to F.
$\circ$	Outer product
$\bullet$	Contracted product
$\otimes$	Kronecker product
$\odot$	Khatri-Rao product
$\circledast$	Hadamard product
$\times_n$	n-mode product.
$*_t$	t-product.

Publications and talks

List of publications

- O. Benchettou, A.H. Bentbib, A. Bouhamidi, K. Kreit, ***Tensorial conditional gradient method for solving multidimensional ill-posed problems***, Applied Numerical Mathematics, 173, 2022, 222-238,

- O. Benchettou, A. H. Bentbib & A. Bouhamidi (2022). ***Tensorial total variation-based image and video restoration with optimized projection methods***, Optimization Methods and Software,

Submitted papers

- O. Benchettou, A.H. Bentbib, A. Bouhamidi, ***An accelerated tensorial double proximal gradient method for total variation regularization problem***. Journal of Optimization Theory and Applications (Under Review).

- O. Benchettou, A.H. Bentbib, A. Bouhamidi, ***An alternating tensorial conditional gradient method for constrained total variation problem*** submitted to Journal *JCAM*.

List of talks

- **Valenciennes Mathematics Seminar**, April 28, 2022, Valenciennes, France.

- **National computer algebra days** held from February 28 to March 4, 2022, Center International de Rencontres Mathématiques (CIRM), Marseille, France.

- **The 5th INTERNATIONAL AEGEAN Symposium on Natural & Medical Sciences**, February 25-26, 2022, Izmir, Turkey.

- **Semaines d'Etude Mathématiques-Entreprises (SEME)**, Institut Polytechnique de Paris, February 14-18, 2022, Palaiseau, France.

- **Istanbul International Modern Scientific Research Congress II** held from December 23 to 25, 2021 in Istanbul, Turkey.
- **AHI EVRAN International Scientific Research Congress** held from November 30 to December 1^{st}, 2021 Kirsehir Ahi Evran University, Turkey.
- **Congress of Young Researchers in Mathematics Applied**, held on October 27-29, 2021 Palaiseau, France.
- **The 9th (Online) International Conference on Applied Analysis and Mathematical Modeling** held from June 11 to 13, 2021 Biruni University Istanbul, Turkey.
- **The EMA seminar of the LMPA laboratory**, June 10, 2021, Calais, France,
- **The International Hybrid Conference on Numerical Analysis and Optimization Days (JANO'13)**, held in Khouribga, Morocco from February 22 to 24, 2021.
- **JST Scientific and Technical Days launched by the Innovations and Technologies Platform (PInTech)**, Hassan II University of Casablanca, from March 22 to 25, 2021.
- **2020 Inter-laboratory Research Day (jril'20)**, held on December 24, 2020, in Fez, Morocco.

CHAPTER 1

Introduction générale en français

Les problèmes inverses existent depuis longtemps dans de nombreuses branches de la physique, des sciences de la terre, de l'ingénierie et des mathématiques. Depuis la naissance de la théorie de l'inversion, le problème inverse avec sa conception de modélisation et d'optimisation devient un sujet multidisciplinaire, qui a reçu beaucoup plus d'attention de nos jours. Sa théorie a été largement développée en raison de l'importance de ses applications telles que la reconstruction d'images, la tomographie, l'identification de paramètres, etc. L'objectif des problèmes inverses, de la conception de modèles et de l'optimisation, est de fournir une meilleure simulation, plus précise et plus efficace dans les applications pratiques.

Étant donné un système avec une entrée (voir Figure 2.1), le problème direct consiste à calculer la sortie. En revanche, étant donné une sortie, le problème inverse consiste à calculer les paramètres du modèle, en tenant compte de l'instabilité de la sortie (dans la plupart des situations, nous disposons de mesures imprécises ou perturbées de la sortie). En d'autres termes, le problème inverse consiste à utiliser les résultats d'observations réelles ou de mesures indirectes pour déduire les autres paramètres caractérisant le système étudié.

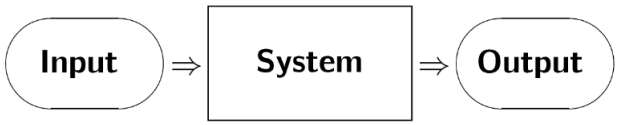


Figure 1.1: Illustration schématique générale d'un problème inverse.

Dans de nombreuses applications techniques et scientifiques, nous rencontrons de grands volumes de données multidimensionnelles, définies sur plusieurs domaines. Par exemple, dans les communications sans fil, les données reçues

par utilisateur peuvent être indexées dans l'espace, le temps et la fréquence [134], ce qui signifie qu'elles sont représentées dans la troisième dimension. De même, dans la reconnaissance faciale [66], nous pouvons associer à chaque personne des conditions d'éclairage, des expressions faciales et une orientation du visage différentes. Et aussi, lorsque l'on traite des données de graphes dans un système de recommandation, les informations résident dans de multiples domaines (par exemple, utilisateurs, films, musique, etc.) [129] ainsi que les tâches de traitement d'images et de vidéos [10, 8].

Pour traiter de tels ensembles de données multidimensionnels, les méthodes classiques consistent à convertir le modèle associé sous une forme bidimensionnelle (forme matricielle). Cependant, avec l'avènement du tenseur algébrique et le développement de ses capacités et caractéristiques dans une classe d'ouvrages intéressants [96, 97, 35], tous les problèmes ci-dessus peuvent être modélisés à l'aide des tenseurs d'ordre supérieur qui se sont avérés extrêmement utiles.

Parmi les points les plus intéressants qui ont alimenté l'intérêt pour la représentation et l'analyse tensorielles, citons la capacité accrue des systèmes de collecte de données à stocker de grands volumes de données multidimensionnelles, ainsi que la modélisation précise qui peut être fournie en laissant les données dans leur forme multidimensionnelle naturelle.

1.1 Problèmes inverses

Commençons par un rappel de la notion de problème inverse dans le contexte linéaire. Nous considérons le problème inverse suivant

$$Au = g, \tag{1.1}$$

où $A : E \to F$ est un opérateur linéaire borné et les ensembles E et F sont des espaces de Banach. Désignons par $\ker(A)$ l'espace nul de l'opérateur A qui est défini par

$$\ker(A) = \{u \in E \mid A(u) = 0\}.$$

Nous appelons l'ensemble $\mathrm{Im}(A)$ le domaine de l'opérateur A, c'est l'image de E sous A, c'est-à-dire $\mathrm{Im}(A) := A(E)$. Conformément à notre définition d'un problème inverse (1.1), nous cherchons une estimation de u à partir de l'observation donnée g. Les problèmes inverses conduisent souvent à des problèmes qui ne satisfont pas à la définition de Hadamard des problèmes bien posés.

Definition 1.1.1 (Problème bien posé au sens de Hadamard [46, 81])**.** *Un problème inverse est bien posé au sens de Hadamard, s'il vérifie les trois conditions suivantes:*

1. *Une solution existe ;*

2. *La solution est unique ;*

3. *La solution dépend de façon continue des données initiales.*

Un problème est dit mal posé si l'une des trois conditions n'est pas vérifiée. Les deux premières conditions de la définition 2.1.1 peuvent être réalisées lorsqu'un opérateur inverse A^{-1} existe, et que son domaine $D(A^{-1})$ (ou l'intervalle $\mathrm{Im}(A)$ de l'opérateur A) coïncide avec E, ce qui n'est rien d'autre que la bijectivité de l'opérateur A. D'autre part, la troisième condition signifie que l'opérateur inverse A^{-1} est continu. Notons que le caractère bien posé du problème inverse peut être défini autrement au sens de Nashed [121] si l'ensemble des observations est un ensemble fermé (le domaine de l'opérateur est fermé).

Dans la littérature, la solution du problème linéaire (1.1) peut être approchée par la méthode des moindres carrés [15]. Cette approche a été proposée indépendamment par Gauss et Legendre au début du 19^{e} siècle. Cette approche a été publié par Legendre, en 1805, sous la forme d'une procédure algébrique d'ajustement d'équations linéaires appliquée à l'astronomie et à la géodésie [104]. Mais en 1809, Carl Friedrich Gauss a la témérité d'affirmer qu'il utilise la méthode depuis 1795, comme méthode de calcul des orbites des corps célestes.

Supposons que les espaces E et F sont des espaces de Hilbert. Les deux points de vue, celui de Gauss et celui de Legendre, s'accordent sur l'idée fondamentale de résoudre le problème (1.1) en minimisant la fonctionnelle quadratique suivantes

$$\Phi(u) = \|Au - g\|_F^2 = \langle A^*Au, u\rangle_E - 2\cdot\langle A^*g, u\rangle_E + \langle g, g\rangle_F, \tag{1.2}$$

où A^* est l'opérateur adjoint de A. Nous désignons par $\langle\cdot,\cdot\rangle_E$ et $\langle\cdot,\cdot\rangle_F$ les produits scalaires dans les espaces E et F, respectivement. Puisque A^*A est une opération définie positive, alors, $\Phi(u)$ est une fonctionnelle convexe. Il est bien connu que pour une fonctionnelle convexe différentiable le problème de trouver

$$\min_u \Phi(u),$$

est équivalent à celui de trouver un point stationnaire, notamment en résolvant une équation de la forme

$$\Phi'(u) = 0.$$

On constate que

$$\begin{cases} \Phi'(u) = 2\,(A^*Au - A^*g), \\ \Phi''(u) = 2A^*A \geq 0. \end{cases} \tag{1.3}$$

Alors, l'équation $\Phi'(u) = 0$ peut se transformer en équation algébrique linéaire

$$A^*Au = A^*g. \tag{1.4}$$

On dit que l'équation (1.4) représente l'équation normale associée au problème principal (1.1). Le nom "équation normale" provient du fait que pour toute solution u, son résidu $Au - g$ est orthogonal (normal) à $\mathrm{Im}(A)$. Pour tout $v \in E$, on a

$$0 = \langle Av, Au - g\rangle_F = \langle v, A^*(Au - g)\rangle_E,$$

ce qui signifie que $Au - g \in \mathrm{Im}(A)^\perp$.

Definition 1.1.2. *[46] Un élément $u \in E$ est appelé*

- *une solution des moindres carrés de (1.1) si*

$$\|Au - g\|_F = \inf\{\|Av - g\|_F,\ v \in E\}.$$

- *une solution de norme minimale (également appelée solution la plus approchée) de (1.1) (est désignée par $u^\dagger$) si*

$$\left\|u^\dagger\right\|_E \leqslant \|v\|_E,$$

pour toutes les solutions des moindres carrés v.

Lemma 1.1.1. *[1] Soient $g \in F$ et l'ensemble $\mathbb{S}$ représente l'ensemble des solutions des moindres carrés. L'ensemble $\mathbb{S}$ est non vide si et seulement si $g \in \mathrm{Im}(A) \oplus \mathrm{Im}(A)^\perp$.*

Puisque $\mathrm{Im}(A)$ n'est pas fermé en général (il n'est jamais fermé pour un opérateur compact, à moins que l'intervalle soit de dimension finie), une solution des moindres carrés peut ne pas exister. Si elle existe, alors la solution de la norme minimale est unique. De plus, la solution de la norme minimale est donc la projection orthogonale de l'élément zéro sur un sous-espace affine défini par

$$\|Au - g\|_F = \min\{\|Av - g\|_F,\ v \in E\}. \tag{1.5}$$

Theorem 1.1.1. *[1] Soit $g \in \mathrm{Im}(A) \oplus \mathrm{Im}(A)^\perp$. Il existe alors une solution unique de norme minimale $u^\dagger$ au problème inverse (1.1) et toutes les solutions des moindres carrés sont données par*

$$\mathbb{S} = \left\{u^\dagger\right\} + \ker(A), \tag{1.6}$$

où $+$ désigne la somme de Minkowski.

Remark 1.1.1. *Dans le cadre d'une dimension finie, il existe toujours une solution des moindres carrés, puisque dans ce cas, le domaine de l'opérateur A est fermé.*

En général, si une solution des moindres carrés existe pour un $g \in F$ donné, alors la solution de la norme minimale peut être calculée à l'aide de l'inverse généralisé de Moore-Penrose de A.

1.2 L'inverse généralisé de Moore-Penrose

L'inverse de Moore-Penrose est défini en théorie des opérateurs en restreignant le domaine et l'étendue de A de telle sorte que l'opérateur restreint résultant soit inversible[46]. En d'autres termes, l'inverse de Moore-Penrose est défini comme suit.

Definition 1.2.1. *[46] Soient $A \in \mathcal{L}(E,F)$ et*

$$\tilde{A} := A|_{\ker(A)^{\perp}} : \ker(A)^{\perp} \to \mathrm{Im}(A)$$

désigne la restriction de A à $\ker(A)^{\perp}$. L'inverse généralisé de Moore-Penrose, désignée par $A^{\dagger}$, est défini comme l'unique extension linéaire de $\tilde{A}^{-1}$ à

$$D\left(A^{\dagger}\right) = \mathrm{Im}(A) \oplus \mathrm{Im}(A)^{\perp},$$

avec

$$\ker\left(A^{\dagger}\right) = \mathrm{Im}(A)^{\perp}.$$

Lorsque le domaine de A est fermé, l'inverse de Moore-Penrose $A^{\dagger}$ est borné [145, 42]. En générale, l'opérateur $A^{\dagger}$ est non borné. Par conséquent, étant donné une observation perturbée g^{δ} telle que $\left\|g^{\delta} - g\right\| \leqslant \delta$, on ne peut pas s'attendre au résultat de convergence suivant

$$A^{\dagger} g^{\delta} \to A^{\dagger} g \text{ as } \delta \to 0. \tag{1.7}$$

Cet inconvénient conduit à un manque d'intérêt pour le concept d'inverse généralisé de Moore-Penrose, et par conséquence, $A^{\dagger} g$ n'est pas une bonne approximation de $A^{\dagger} g^{\delta}$ en raison du caractère non borné de $A^{\dagger}$, même s'il existe. Ainsi, pour assurer la convergence, nous remplaçons l'opérateur $A^{\dagger}$ par une famille d'opérateurs bien posés (bornés) R_{μ} satisfaits

$$R_{\mu}\left(g^{\delta}\right) \underset{\delta \to 0}{\longrightarrow} A^{\dagger} g,\ \forall g \in D\left(A^{\dagger}\right),\ g^{\delta} \in F,\ \ s.t.\ \left\|g - g^{\delta}\right\|_{F} \leqslant \delta, \tag{1.8}$$

où $\mu = \mu\left(\delta, g^{\delta}\right)$. La famille $\left\{R_{\mu}\right\}_{\mu > 0}$ d'opérateurs continus est appelée opérateurs de régularisation de $A^{\dagger}$. En terme générale, la régularisation est l'approximation d'un problème mal posé par une famille de problèmes voisins bien posés, comme

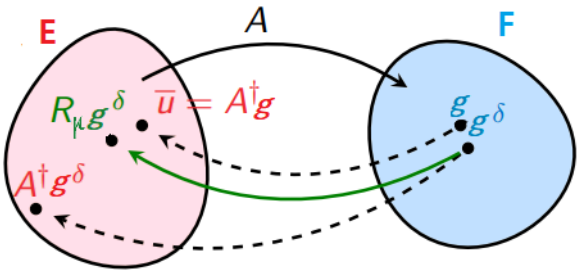


Figure 1.2: Illustration de l'instabilité de l'inverse de Moore-Penrose.

l'illustre la figure 1.2. Elle est définie par deux éléments essentiels: L'opérateur de régularisation R et le paramètre de régularisation μ. Le choix de ces deux éléments influence l'efficacité de l'approximation.

En 1963, le mathématicien russe Andrey Nikolayevich Tikhonov a formulé une définition célèbre de l'algorithme de régularisation qui est une conception de base dans la théorie moderne des problèmes mal posés.

Definition 1.2.2. *[146] Un opérateur R_μ est appelé opérateur de régularisation $R_\mu = R\left(\mu, g^\delta\right) \equiv R_\mu\left(g^\delta\right)$ s'il possède deux propriétés :*

1. *L'opérateur R est défini de $(0,+\infty) \times F$ à E, pour tout $\mu > 0$ et $g^\delta \in F$.*
2. *Pour tout $u \in E$ et $g^\delta \in F$ tel que $Au = g$ et $\left\|g - g^\delta\right\| \leq \delta$, on a*

$$u_\mu^\delta = R_\mu\left(g^\delta\right) \to u \text{ as } \delta \to 0. \tag{1.9}$$

où u_μ^δ est une solution régularisée du problème inverse (1.1).

Un problème est dit régularisable s'il existe au moins un opérateur de régularisation. En général, tous les problèmes mathématiques peuvent être répartis dans les classes suivantes:

1. Problèmes bien posés;
2. Problèmes mal posés régularisables;
3. Problèmes mal posés non régularisables.

Tous les problèmes bien posés sont régularisables, car on peut prendre $R_\mu = A^{-1}$ où la connaissance de $\mu > 0$ n'est pas obligatoire dans ce cas. En revanche, la connaissance du paramètre de régularisation μ est cruciale si notre problème est mal posé.

Considérons une régularisation linéaire où nous pouvons diviser l'erreur totale entre la solution régularisée du problème perturbé $R_\mu g^\delta$ et la solution de norme minimale du problème $u^\dagger = A^\dagger g$ comme suit

$$\underbrace{\left\|R_\mu g^\delta - u^\dagger\right\|_E}_{\text{erreur totale}} \leqslant \underbrace{\delta\left\|R_\mu\right\|_{\mathcal{L}(F,E)}}_{\text{erreur de données}} + \underbrace{\left\|R_\mu g - A^\dagger g\right\|_E}_{\text{erreur d'approximation}} \tag{1.10}$$

Ceci peut également être illustré comme dans Figure 1.3. Figure 1.3 montre que pour un petit paramètre de régularisation μ, l'erreur dans les données est amplifiée par le caractère mal posé du problème et pour un grand paramètre μ, l'opérateur R_μ est une mauvaise approximation de l'inverse de Moore-Penrose. Cela signifie que le choix du paramètre de régularisation contrôle le compromis entre l'approximation et la stabilité. En d'autres termes, le paramètre de régularisation équilibre l'importance respective entre les deux termes dans (1.10).

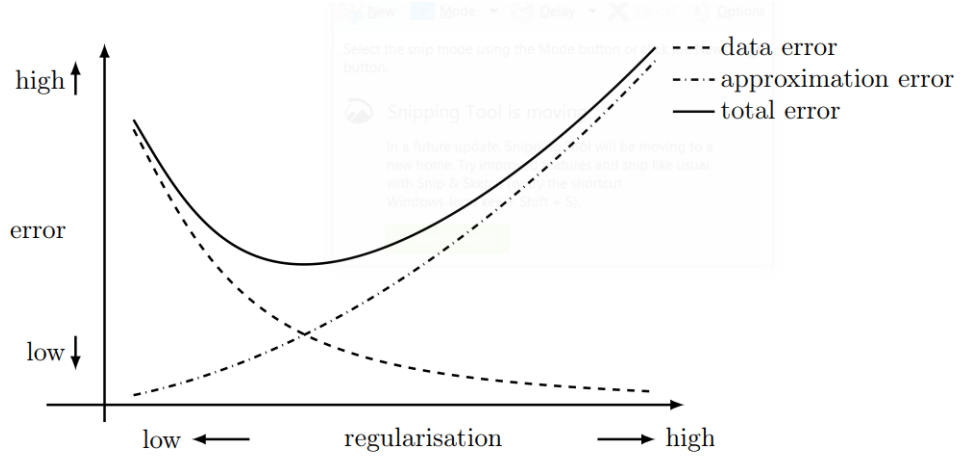


Figure 1.3: L'erreur totale entre une solution régularisée et la solution de norme minimale se décompose en erreur de données et erreur d'approximation.

Notez que l'erreur de données définie comme $\delta\left\|R_\mu\right\|_{\mathcal{L}(F,E)}$ ne reste pas bornée pour $\mu \longrightarrow 0$. Cependant, l'erreur d'approximation $\left\|R_\mu g - A^\dagger g\right\|_E$ s'annule pour $\mu \longrightarrow 0$, à cause de la convergence ponctuelle de R_μ vers $A^\dagger$.

1.3 Problématique et enjeux

Dans la littérature, la modélisation d'un problème linéaire peut se faire en utilisant l'algèbre matricielle, en partant d'un modèle physique et en arrivant à une équation matricielle à résoudre. Même dans les cas où le problème est dans une dimension supérieure, les méthodes classiques consistent à projeter le problème en 2D en utilisant le processus de vectorisation afin de fournir un modèle matriciel. Aujourd'hui, compte tenu du développement de l'algèbre tensorielle, nous pouvons modéliser et résoudre des problèmes dans des dimensions supérieures en utilisant des opérations tensorielles [97, 35, 96]. Le principal avantage de la représentation et de l'analyse tensorielle est la capacité de traiter d'énormes volumes de données, ainsi que de maintenir la nature multidimensionnelle du problème à résoudre.

Contrairement au cas des matrices, l'algèbre tensorielle offre différents produits et décompositions selon l'ordre, la dimension et la similarité de structure entre les éléments du modèle, ce qui rend la modélisation en haute dimension plus complexe. L'objectif principal de cette thèse était la représentation et la régularisation de problèmes inverses de dimension supérieure en utilisant l'algèbre tensorielle.

Le plan de cette thèse est présenté comme suit. Le chapitre 3 donnera un état de l'art sur la forme variationnelle de la régularisation, y compris le caractère bien posé du problème général de minimisation, certains choix du régularisateur ainsi que les méthodes de sélection des paramètres de régularisation. Le chapitre 4 est consacré à la terminologie de l'algèbre tensorielle qui servira à la généralisation et à la modélisation de problèmes en dimension supérieure. Dans le chapitre 5, le problème de régularisation tensorielle de Tikhonov est résolu en étudiant différents cas, soit le cas du problème sans contraint passant par les méthodes de type sous-espace de Krylov, soit dans le cas sous contraint en développant l'algorithme du gradient conditionnel. Le problème de la variation totale tensorielle sans contrainte est étudié dans le chapitre 6 où les techniques de projection sont utilisées pour résoudre les sous-problèmes issus de la méthode des directions alternée (*ADMM*). D'autre part, dans le chapitre 7, le problème de la variation totale tensorielle sous contrainte convexe a été résolue en développant des approches de type gradient. Pour illustrer l'efficacité des approches développées, une application au traitement d'images et de vidéos sera donnée dans le chapitre 8. Enfin, nous énonçons les conclusions et les perspectives dans le chapitre 9.

CHAPTER 2

General Introduction

The field of inverse problems has existed in many branches of physics, earth science, engineering, and mathematics for a long time. From the beginning of the birth of the inversion theory, the inverse problem with its modeling design and optimization becomes a multi-disciplinary subject, which has received much more attention nowadays. Its theory has been widely developed due to the importance of its applications such as in image reconstruction, tomography, parameter identification, etc. All of which require powerful computers and reliable solution methods to perform the calculation. Therefore, the aim of the inverse problems, modeling design, and optimization is to provide a more accurate, and more efficient simulation in practical applications.

Given a system with an known input, the forward problem consists of computing the output (Figure 2.1). In contrast, the inverse problem consists of computing either the input or the system, given the other two quantities taking into account that in most situations, we have imprecise (disturbed) measurements of the output. In other words, the inverse problem consists in using the results of actual observations or indirect measurements to infer the model or the values of the parameters characterizing the system under investigation.

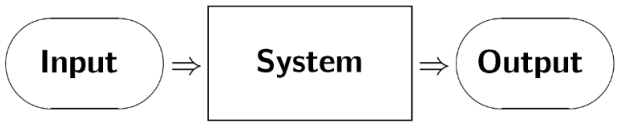


Figure 2.1: A general schematic illustration of an inverse problem.

In many engineering and scientific applications, we encounter large volumes of multidimensional data, defined over multiple domains. For example, in wireless communications, received data per user may be indexed in space, time, and

frequency [134], which means it is represented in the third dimension. Similarly, in facial recognition [66], we may associate with each person different lighting conditions, facial expressions, and face orientation. And also, when dealing with graph data in a recommender system, information resides on multiple domains (e.g., users, movies, music, and so on)[129] as well as image and video processing tasks [10, 8]. To process such multidimensional datasets, classical methods consist in converting the associated model into a two-dimensional form (matrix form). However, with the advent of the algebraic tensor and the development of its capabilities and characteristics in a class of interesting works [96, 97, 35], all of the above problems can be modeled using higher-order tensors that have proven to be extremely useful.

Among the most interesting points that have fueled interest in tensor representation and analysis are the increased ability of data collection systems to store large volumes of multidimensional data, as well as the accurate modeling that can be provided by leaving the data in its natural, multidimensional form.

2.1 Inverse problems

Let us start with a reminder of the notion of inverse problem in the linear context. Considering the inverse problem

$$Au = g, \tag{2.1}$$

where $A : E \to F$ is a linear bounded operator and the sets E and F are Banach spaces. Let us denote by $\ker(A)$ the null space of the operator A which is defined by

$$\ker(A) = \{u \in E \mid A(u) = 0\}.$$

We call the set $\mathrm{Im}(A)$ the range of the operator A, it is the image of E under A, that is, $\mathrm{Im}(A) := A(E)$. In accordance with our definition of an inverse problem, in (2.1), we seek an estimate of u starting by the given observation g.

Inverse problems frequently lead to mathematical problems that do not fulfill Hadamard's definition of well-posedness which is defined as follows.

Definition 2.1.1 (Hadamard well-posed problem [46, 81])**.** *An inverse problem is well-posed in the sense of Hadamard, if it verifies the three conditions:*

1. *A solution exists,*
2. *The solution is unique,*
3. *The solution depends continuously on the initial data.*

A problem is called ill-posed if one of the three conditions is not verified. The first two conditions of Definition 2.1.1 can be reached when an inverse operator A^{-1} exists, and its domain $D(A^{-1})$ (or the range $\mathrm{Im}(A)$ of operator A) coincides with E, which is nothing but the bijectivity of the operator A. On the other hand, the third condition means that the inverse operator A^{-1} is continuous.

Remark 2.1.1. *The well-Posedness of the inverse problem can be defined otherwise in the sense of Nashed [121] if the set of observations is a closed set(the range of the operator is closed).*

In the literature, the solution of the linear problem (2.1) can be approximated by the least square method [15]. This approach was proposed independently by Gauss and Legendre in the beginning of the 19^{th} century. The first clear and concise exposition of the method of least squares was published by Legendre, in 1805, as an algebraic procedure for fitting linear equations applied in astronomy and geodesy [104]. But in 1809, Carl Friedrich Gauss had the temerity to claim that he had been using the method since 1795, as method of calculating the orbits of celestial bodies.

In the rest of the current chapter, let us suppose that the spaces E and F are Hilbert spaces. We denote the inner products in these spaces $\langle\cdot,\cdot\rangle_E$ and $\langle\cdot,\cdot\rangle_F$, respectively. Both points of view, Gauss's and Legendre's, agree on the basic idea of solving the problem (2.1) by minimizing the quadratic functional

$$\Phi(u) = \|Au - g\|_F^2 = \langle A^*Au, u\rangle_E - 2\cdot\langle A^*g, u\rangle_E + \langle g, g\rangle_F, \tag{2.2}$$

where A^* is adjoint operator of A . Since A^*A is positive-definite operation, then, $\Phi(u)$ is a convex functional. It is well-known that for a convex differentiable functional the problem to find

$$\min_u \Phi(u),$$

is equivalent to that to find a stationary point, notably solving an equation of the form

$$\Phi'(u) = 0.$$

It is easy to see that

$$\begin{cases} \Phi'(u) = 2\cdot(A^*Au - A^*g), \\ \Phi''(u) = 2\cdot A^*A \geq 0. \end{cases} \tag{2.3}$$

Then, the equation $\Phi'(u) = 0$, which means that the gradient of the discrepancy is equal to zero, is turned into the linear algebraic equation

$$A^*Au = A^*g. \tag{2.4}$$

We say that the equation (2.4) represents the normal equation associated with the main problem (2.1). The name "normal" is derived from the fact that for any solution u, its residual $Au - g$ is orthogonal (normal) to $\mathrm{Im}(A)$. This can be readily seen, as we have for any $v \in E$ that

$$0 = \langle Av, Au - g\rangle_F = \langle v, A^*(Au - g)\rangle_E,$$

which means that $Au - g \in \mathrm{Im}(A)^\perp$.

Definition 2.1.2. *[46] An element $u \in E$ is called*

- *a least squares solution of (2.1) if*

$$\|Au - g\|_F = \inf\{\|Av - g\|_F,\ v \in E\}.$$

- *a minimal-norm solution (also called the best-approximate solution)of (2.1) (and is denoted by $u^\dagger$) if*

$$\left\|u^\dagger\right\|_E \leqslant \|v\|_E,$$

for all least squares solutions v.

Lemma 2.1.1. *[90, 1] Let $g \in F$. Let the set $\mathbb{S}$ stands for the set of least squares solution. The set $\mathbb{S}$ is non-empty if and only if $g \in \mathrm{Im}(A) \oplus \mathrm{Im}(A)^\perp$.*

Since $\mathrm{Im}(A)$ is not closed in general (it is never closed for a compact operator, unless the range is of finite dimension), a least squares solution may not exist. If it exists, then the minimal-norm solution is unique. Moreover, the minimal-norm solution is the orthogonal projection of the zero element onto an affine subspace defined by

$$\|Au - g\|_F = \min\{\|Av - g\|_F,\ v \in E\}. \tag{2.5}$$

Theorem 2.1.1. *[90, 1] Let $g \in \mathrm{Im}(A) \oplus \mathrm{Im}(A)^\perp$. Then there exists a unique minimal norm solution $u^\dagger$ to the inverse problem (2.1) and all least squares solutions are given by*

$$\mathbb{S} = \left\{u^\dagger\right\} + \ker(A), \tag{2.6}$$

where + denotes the Minkowski sum.

Remark 2.1.2. *In the finite dimensional setting, there always exists a least squares solution, since in this case, the range of A is closed.*

In general, if a least-squares solution exists for a given $g \in F$, then the minimal-norm solution can be computed (at least in theory) using the Moore-penrose generalized inverse of A, which will turn out to be the solution operator mapping g onto the best-approximate solution of (2.1).

2.2 Moore-Penrose inverse and the need of regularization

The Moore-Penrose inverse is defined in an operator-theoretic way by restricting the domain and range of A in such a way that the resulting restricted operator is invertible, and its inverse will then be extended to its maximal domain [46]. In other words, the Moore-Penrose inverse is defined as follows.

Definition 2.2.1. *[46] Let $A \in \mathcal{L}(E,F)$ and let*

$$\tilde{A} := A|_{\ker(A)^\perp} : \ker(A)^\perp \to \mathrm{Im}(A)$$

denotes the restriction of A to $\ker(A)^\perp$. The Moore-Penrose inverse $A^\dagger$ is defined as the unique linear extension of $\tilde{A}^{-1}$ to

$$D\left(A^\dagger\right) = \mathrm{Im}(A) \oplus \mathrm{Im}(A)^\perp,$$

with

$$\ker\left(A^\dagger\right) = \mathrm{Im}(A)^\perp.$$

A particular case when the operator A has furthermore a closed range, the Moore-Penrose inverse $A^\dagger$ is bounded [145, 42]. Otherwise, $A^\dagger$ is unbounded. Therefore, given a disturbed data g^δ such that $\left\|g^\delta - g\right\| \leqslant \delta$, we cannot expect the convergence result

$$A^\dagger g^\delta \to A^\dagger g \text{ as } \delta \to 0. \tag{2.7}$$

This drawback leads to a lack of interest in the concept of Moore-Penrose generalized inverse. As a consequence, $A^\dagger g$ is certainly not a good approximation of $A^\dagger g^\delta$ due to the unboundedness of $A^\dagger$ even if it exists. Thus, to ensure the convergence, we replace the operator $A^\dagger$ with a family of well-posed (bounded) operators R_μ satisfied

$$R_\mu\left(g^\delta\right) \underset{\delta\to 0}{\longrightarrow} A^\dagger g,\ \forall g \in D\left(A^\dagger\right),\ g^\delta \in F,\ \ s.t.\ \left\|g - g^\delta\right\|_F \leqslant \delta, \tag{2.8}$$

where $\mu = \mu\left(\delta, g^\delta\right)$. The family $\left\{R_\mu\right\}_{\mu>0}$ of continuous operators is called regularization operators of $A^\dagger$. In general terms, regularization is the approximation of an ill-posed problem by a family of neighbouring well-posed problems, as illustrated in Figure 2.2. This regularized approximation is defined by two essential elements: The regularization operator R and the regularization parameter μ. The choice of these two elements influences the efficiency of the approximation.

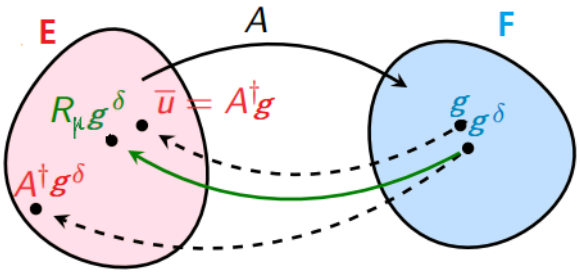


Figure 2.2: Illustration of the instability of the Moore-penrose inverse.

In 1963, the Russian mathematician, Andrey Nikolayevich Tikhonov formulated a famous definition of the regularizing algorithm that is a basic conception in the modern theory of ill-posed problems. A classical definition was expressed as follows.

Definition 2.2.2. *[146] An operator R_μ is called a regularizing operator $R_\mu = R\left(\mu, g^\delta\right) \equiv R_\mu\left(g^\delta\right)$ if it is possessing two properties:*

1. *R_μ is defined for any $\mu > 0$, $g^\delta \in F$, and is mapping $(0, +\infty) \times F$ into E.*
2. *For any $u \in E$ and for any $g^\delta \in F$ such that $Au = g$ and $\left\|g - g^\delta\right\| \leq \delta$, we have*

$$u_\mu^\delta = R_\mu\left(g^\delta\right) \to u \text{ as } \delta \to 0. \tag{2.9}$$

where u_μ^δ is the regularized solution of (2.1).

After the efficiency of this concept, regulation has seen significant growth by developing different definitions of this process depending on the framework and field of work. A problem of solving an operator equation is called regularizable if there exists at least one regularizing operator. At the present time, all mathematical problems can be divided into the following classes:

1. well-posed problems;
2. ill-posed regularizable problems;
3. ill-posed nonregularizable problems.

All well-posed problems are regularizable as it can be taken $R_\mu = A^{-1}$ where the knowledge of the regularization parameter $\mu > 0$ is not obligatory in this case. Otherwise, the knowledge of the parameter μ which means the knowledge of the error δ is crucial if our problem is ill-posed.

Let us consider a linear regularization where we can split the total error between the regularized solution of the disturbed problem $R_\mu g^\delta$ and the minimal norm solution of the problem $u^\dagger = A^\dagger g$ as follows

$$\underbrace{\left\|R_\mu g^\delta - u^\dagger\right\|_E}_{\text{total error}} \leqslant \underbrace{\delta\left\|R_\mu\right\|_{\mathcal{L}(F,E)}}_{\text{data error}} + \underbrace{\left\|R_\mu g - A^\dagger g\right\|_E}_{\text{approximation error}} \tag{2.10}$$

This can also be illustrated as in Figure 2.3. Figure 2.3 shows that for a small regularization parameter μ the error in the data gets amplified through the ill-posedness of the problem and for large μ the operator R_μ is a poor approximation of the Moore–Penrose inverse. Which means that the choice of the regularization parameter controls the trade-off between approximation and stability. In other words, the regularization parameter balances the respective importance between the two terms in (2.10).

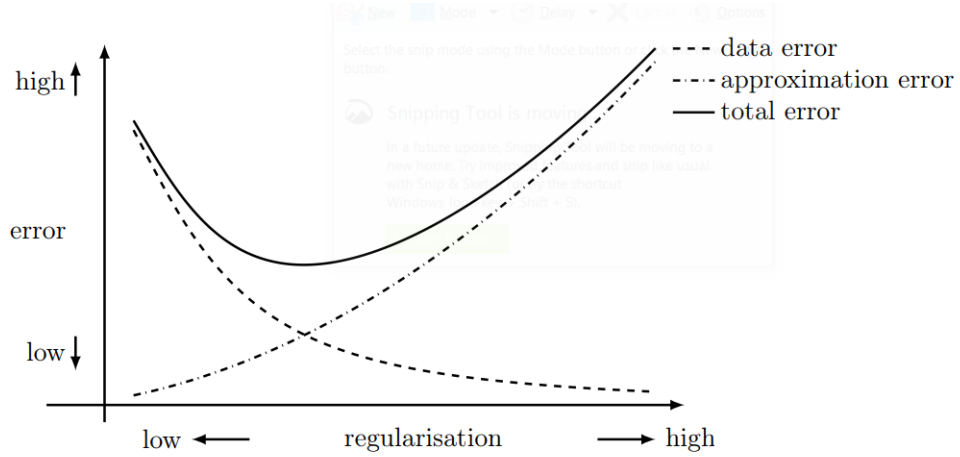


Figure 2.3: The total error between a regularized solution and the minimal norm solution decomposes into the data error and the approximation error.

Note that the data error defined as $\delta\left\|R_\mu\right\|_{\mathcal{L}(F,E)}$ does not stay bounded for $\mu \longrightarrow 0$. However, the approximation error $\left\|R_\mu g - A^\dagger g\right\|_E$ vanishes for $\mu \longrightarrow 0$, due to the pointwise convergence of R_μ to $A^\dagger$. Hence it becomes evident from (2.10) that a good choice of μ depends on δ, in addition, it needs to be chosen such that the approximation error becomes as small as possible, whilst the data error is being kept at bay.

2.3 Inverse source problem

Inverse problems can be found in many scientific fields. In this section, we will give some well-known examples like the image degradation model and the discrete computed tomography problem.

2.3.1 Image deblurring

Image deblurring is the process of reconstructing a digital image from a blurred and noisy one, based on a mathematical model of the blurring operation. In the continuous setting, image deblurring is considered as a first-kind Fredholm integral equation of the generic form

$$\int_0^1 \int_0^1 k(s,t) f(t) dt_1 dt_2 = g(s), \quad s \in [0,1] \times [0,1], \tag{2.11}$$

in which the two functions f and g, that represent the sharp and blurred images, are both functions of two spatial variables $s = (s_1, s_2)$ and $t = (t_1, t_2)$. Let the two $N \times N$ arrays X and B represent the unknown, sharp image and the recorded blurred and noisy image, respectively, and let $X_{i,j}$ and $B_{i,j}$ denote the array elements, also known as pixels. Let the vectors x and b, of length $n = N^2$, consist of the vectorized (stacked) columns of X and B, i.e.,

$$\left.\begin{array}{l} x_\ell = X_{ij} \\ b_\ell = B_{ij} \end{array}\right\}, \text{ where } \ell = (j-1)N + i, \tag{2.12}$$

where the operation described by (2.12) called the vectorization of a matrix, denoted by ***vec*** ($x = \boldsymbol{vec}(X)$ and $b = \boldsymbol{vec}(B)$). Then a good model for the blurring is very often of the form $Ax = b$, where A is an $N^2 \times N^2$ given matrix, known also as the blurring matrix. The construction of the blurring matrix depends on the two essential ingredients the point spread function (PSF) that defines how each pixel is blurred and the boundary conditions that specify our assumption on the scene just outside our image[74, 89, 70]. Therefore, the process of restoring an image amounts to solving an inverse problem of the form (2.1).

2.3.2 Computed Tomography Imaging

In general, tomography can be defined as the creation of an image from the projection data associated with cross-sectional scans of an object[120, 93, 73]. The process can be described as follows. Suppose we apply multiple projections to an object. In that case, all these projections will pass through the object and

fall on a screen called a sensor, which will provide projection curves containing the primary information used as an observation in our inverse problem (see Figure 2.4).

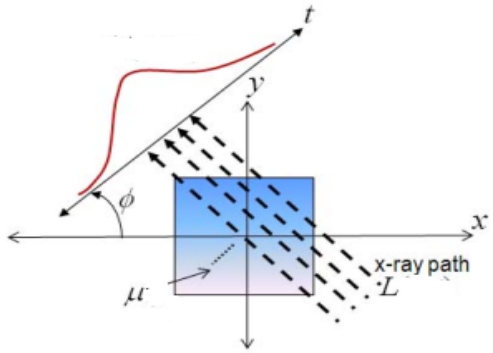


Figure 2.4: The data acquisition process during scanning.

From a physical point of view, the computed tomography model is expressed using intensities. As the high-energy photons pass through a material, their energy is decreasing which is known as attenuation or absorption. This phenomenon can be expressed, in the case of a poly-energetic x-ray source using the spectral version of Beer's law, by

$$\Lambda_i(E) = \sigma_i(E)\exp\left(-\int_{L_i}\mu(r,E)\mathrm{d}l\right), \quad i = 1,\dots,M,$$

where M is the total number of rays, E is the energy, $r \in \mathbb{R}^3$ defines the spatial position, μ is the energy-dependent attenuation coefficient, σ_i is the energy-dependent intensity flux of the x-ray source associated with ray i, and $\Lambda_i(E)$ is the spectrum of the x-ray beam incident on the detector[137, 73, 93].

By introducing the following parameterization

$$\mu(r,E) = \sum_{j=1}^{N}\mu_j(E)\chi_j(r),$$

where N denotes the number of pixels, and by defining $A_{ij} = \int_{L_i}\chi_j(r)\mathrm{d}l$, we arrive at the element-wise model

$$\Lambda_i(E_k) = \sigma_i(E_k)\exp\left(-\sum_{j=1}^{N}A_{ij}\mu_j(E_k)\right), \quad i = 1,\dots,M, \quad k = 1,\dots,K. \tag{2.13}$$

For basis functions $\chi_j(r)$, we use a pixel basis such that A_{ij} is the length of the intersection of the ith ray and the jth pixel and K denotes the number of the consider separate channels corresponding to K energies $E_1, \ldots, E_K$.

For photon counting detectors, it is common to assume that the measurements are Poisson distributed with a parameter $\Lambda(E)$. This assumption leads to the measurement model

$$Y_{ik} \sim \text{Poiss}\{\Lambda_i(E_k)\}, \quad i = 1, \ldots, M,\ k = 1, \ldots, K.$$

If we define $X_{jk} = \mu_j(E_k)$ and $S_{ik} = \sigma_i(E_k)$, we get the discrete linear model

$$B_{ik} = -\ln\left(\frac{Y_{ik}}{S_{ik}}\right) = \sum_{j=1}^{N} A_{ij} X_{jk},$$

which is clearly leads to a system of linear equations

$$b = \tilde{A}x,$$

where $x = \boldsymbol{vec}(X) \in \mathbb{R}^{NK}$, $b = \text{vec}(B)$ and the matrix $\tilde{A} = I_{K\times K} \otimes A$ denotes the projection matrix.

2.4 Problematic & outline

In the literature, modeling some linear problem can be done using matrix algebra, as in the examples given in the previous section, starting with a physical model and arriving at a matrix equation to solve. Even in the cases where the problem is in a higher dimension, the classical methods consist in projecting the problem in 2D using the vectorization process in order to provide a matrix model. Nowadays, in view of the development of tensor algebra, we can model and solve problems in higher dimensions by using tensor operations [97, 35, 96]. The main purpose of using tensor representation and analysis is the ability to treat huge data volume, as well as, maintain the multidimensional nature of the problem to be solved.

Unlike the matrix case, the tensor algebra offers different products and decompositions depending on the order, the dimension, and the structure similarity between the elements of the model which make the modeling in high dimension more complex. The main goal of this thesis was the representation and regularization of higher dimensional inverse problems using tensor algebra.

The outline of this thesis is presented as follows. Chapter 3 will give a state of art on the variational form of regularization including the well-posedness

of the general minimization problem, some choices of the regularizer as well as the regularization parameter selection methods. Chapter 4 is devoted to the tensor algebra terminology that will serve in the generalization and the modeling of high dimensional problem. In Chapter 5, the tensorial Tikhonov regularization problem is solved by investigating different cases, either the case of the unconstrained problem passing by the Krylov subspace method or in the constrained case by developing the conditional gradient algorithm. The problem of unconstrained tensorial total variation is investigated in the Chapter 6 where the projection techniques are using to solve the subproblems stemmed from the alternating direction method for multipliers (*ADMM*). On the other hand, in Chapter 7, the tensorial total variation under a convex constraint has been solved by developing gradient-like approaches. To illustrate the efficiency of the developed approaches, an application to image and video processing will be given in Chapter 8. Finally, we state the conclusions and the perspectives in Chapter 9.

CHAPTER 3

Variational form of regularization

3.1 Introduction

The regularization can be represented using different forms. The spectral regularization can be considered as the simplest one to treat. It is based on the spectral decomposition of the operator in the inverse problem [74]. The main idea behind the spectral decomposition is discarding the small singular value that amplify the error during the inversion process. The resulting method is referred to as the truncated *SVD* [72]. However, the most widely referenced regularization method is of variational form[79, 146].

The variational form of regularization consists in solving a nearby well-posed optimization problem and take its minimizer, denoted by $u_\mu^\delta : \Omega \longrightarrow \mathbb{R}$, as a solution. The variational regularization problem is given by

$$\min_{u\in\mathcal{K}} \Big(\mathcal{J}_\mu(u) := \|Au - g^\delta\|^p + \mu\mathcal{R}(u) \Big). \tag{3.1}$$

The objective functional of this minimization problem consists of two terms, the fidelity term $\|Au - g^\delta\|^p$, which can be replaced by another fidelity term given in Table 3.1 depending on the situation. The regularization term $\mu\mathcal{R}(u)$ which consists of a regularization parameter μ and a regularization operator $\mathcal{R}$. In addition to the convex constraint $\mathcal{K}$.

The fidelity term measures the likelihood of any u based on knowledge of the observed data g^δ and the observation process A. On the other hand, the regularization operator $\mathcal{R}$ promotes the regularization effect on the solution. While, the regularization positive parameter μ is used to balance the respective importance of the two terms in the minimization problem. The choice of these two terms has received much attention in the literature, so that the choice of

the first depends on the structure of the perturbation (see Table 3.1), and the choice of the second depends on the structure of the solution (smooth, sparse, with discontinuities, etc.), see Table 3.2 for further choices of the regularization operator.

Noise model	Fidelity term
Additive Gaussian	$\left\|Au - g^\delta\right\|_{L^2}^2$
Additive impulsive	$\left\|Au - g^\delta\right\|_{L^1}$
Poisson	$\int_\Omega \left(Au - g^\delta \log Au\right) dx$
Speckle noise	$\int_\Omega \left(\log Au + \frac{g^\delta}{Au}\right) dx$
Huber	$\left\|L_\delta\left(Au - g^\delta\right)\right\|_{L^1}, L_\delta(t) = \begin{cases} \frac{1}{2}t^2, & \|t\| \le \delta, \\ \delta\left(t - \frac{\delta}{2}\right), & \|t\| > \delta. \end{cases}$

Table 3.1: Common fidelity functionals for different error structure.

Prior model	The regularization operator $\mathcal{R}(u)$
Generalized Gaussian	$\|u\|_{L^p}^p$, $1 \le \nu \le 2$
Total variation (TV) or (BV)	$\|u\|_{BV}$ or $\|u\|_{\mathrm{TV}}^p$
Sobolev	$\|u\|_{W^{1,p}}$
Elastic-net	$\|u\|_{\ell^1} + \frac{\gamma}{2}\|u\|_{\ell^2}^2$
Sparsity	$\|u\|_{\ell^p}^p$, $0 \le p < 2$

Table 3.2: Common regularization operators.

The structure of the variational form of regularization comes back to the Bayesian inference which is defined by the fundamental relation given by the Bayes rule

$$P(U \mid V) = \frac{P(V \mid U)P(U)}{P(V)}, \tag{3.2}$$

where U and B are two events, $P(U)$ and $P(V)$ are the probabilities of the event U and B, respectively, with $P(V) \neq 0$ and $P(U \mid V)$ is the probability of U given V, also known as the conditional probability. For two continuous random variables X and Y, Bayes theorem may be similarly derived from the definition of conditional density as follow:

$$f_{X|Y=y}(x) = \frac{f_{Y|X=x}(y) f_X(x)}{f_Y(y)}. \tag{3.3}$$

Thus, by considering the elements u and g^δ of our ill-posed problem $Au = g^\delta$ as

continuous random variables, the Bayes rule can be reformulated to the following relation

$$f_{u|g^\delta} f_{g^\delta} = f_{g^\delta|u} f_u. \tag{3.4}$$

In the simplest situations, when we consider the additive Gaussian noise with average 0 and standard deviation σ, the density for the probability of u knowing g^δ will be given by

$$f_{u|g^\delta} = Z(g^\delta) e^{-p(u) - \frac{1}{2\sigma^2} \sum_{i,j} |g^\delta_{i,j} - (Au)_{i,j}|^2}, \tag{3.5}$$

where $p(u)$ is the priori probability density and $Z(g^\delta)$ denotes a renormalization factor. The idea of maximum a posteriori (MAP) reconstruction [113] is to find the best approximation u as the one which maximize this probability which is equivalent to solve the following discrete minimization problem

$$\min_u \left(p(u) + \frac{1}{2\sigma^2} \sum_{i,j} |g^\delta_{i,j} - (Au)_{i,j}|^2 \right), \tag{3.6}$$

which is nothing but the variational regularization model (3.1) with $p = 2$, namely,

$$\min_u \left(\mu \mathcal{R}(u) + \|Au - g^\delta\|_2^2 \right), \tag{3.7}$$

where the operation $\mathcal{R}$ is a functional corresponding to the a priori probability density $p(u)$, and the scalar $\mu > 0$ denotes a weight balancing the respective importance of the two terms in the problem.

3.2 Well-posedness of the variational regularization problem

The well-posedness of the constrained minimization problem (3.1) will be studied by the verification of three characteristics which are the existence, the stability and the consistency[63, 79].

3.2.1 Existence of a minimum

Theorem 3.2.1. *[79] Let E be reflexive and the set $\mathcal{K} \subset E$ be convex and closed. Suppose that*

1. *The functional $\mathcal{J}_\mu$ is coercive, i.e., for any sequence $(u_n)_n$ such that the functional value $\left(\mathcal{J}_\mu(u_n)\right)$ is uniformly bounded, then, the sequence $(u_n)_n$ is uniformly*

bounded in E.

2. *The functional $\mathcal{R}$ is sequentially weak lower semi-continuous.*

3. *The operator A is bounded.*

Then for every $\mu > 0$, there exists at least one minimizer u_μ^δ to the functional $\mathcal{J}_\mu$ defined by (3.1).

Proof. Since the functional $\mathcal{J}_\mu$ is nonnegative, there exists a minimizing sequence $(u_n)_n \subset \mathcal{K}$, such that

$$\lim_{n\to\infty} \mathcal{J}_\mu(u_n) = \inf_{u\in\mathcal{K}} \mathcal{J}_\mu(u) := \eta.$$

Hence, the sequence of functional values $\left(\mathcal{J}_\mu(u_n)\right)_n$ is uniformly bounded. By the coercivity of the cost function, there exists a subsequence of $(u_n)_n$, also denoted by $(u_n)_n$, and there exist $u^* \in E$ such that $(u_n)_n$ converges weakly to u^*. In addition, by the closedness and convexity of the set $\mathcal{K}$, it follows that $u^* \in \mathcal{K}$. On the other hand, from the lower semi-continuity of the operator $\mathcal{R}$ and the weak lower semicontinuity of norms, we deduce

$$\begin{aligned}\left\|Au^* - g^\delta\right\|^p + \mu\mathcal{R}(u^*) &\le \liminf_{n\to\infty} \left\|Au_n - g^\delta\right\|^p + \mu \liminf_{n\to\infty} \mathcal{R}(u_n) \\ &\le \liminf_{n\to\infty} \left(\left\|Au_n - g^\delta\right\|^p + \mu\mathcal{R}(u_n)\right) = \eta\end{aligned}$$

i.e., u^* is a minimizer to $\mathcal{J}_\mu$. □

Remark 3.2.1. *In general, the existence result holds if the chosen fidelity functional is weakly semicontinuous lower with respect to u, for any $g^\delta \in F$ fixed.*

3.2.2 Stability of a minimum

Let us first recall the H-property of a functional K [105].

Definition 3.2.1. *A functional $K : E \longrightarrow F$ is said to have the H-property on the space E if any sequence $(u_n)_n \subset E$ that satisfies that the conditions $u_n \to u$ weakly for some $u \in E$ and $K(u_n) \to K(u)$ imply that u_n converges strongly to u in E.*

Remark 3.2.2. *Norms on Hilbert spaces, L^p spaces and Sobolev spaces $W^{m,p}$ with $1 < p < \infty$ and $m \ge 1$ satisfy the H-property.*

Now we state the stability result.

Theorem 3.2.2. *[79] Let the three assumption in Theorem 3.2.1 hold. Let $(g_n)_n \subset F$ be the sequence that convergent to g^δ in F, and let $(u_n)_n$ be a minimizing sequence of the functional $\mathcal{J}_\mu$ with g_n in place of g^δ. Then, there exist a subsequence of $(u_n)_n$ converge weakly to a minimizer to $\mathcal{J}_\mu$. Further, if the minimizer to $\mathcal{J}_\mu$ is unique, then the whole sequence converges weakly. Moreover, the strong convergence verifies, if the functional $\mathcal{R}$ satisfies the H-property.*

Proof. Let u_μ^δ be a minimizer to $\mathcal{J}_\mu$. The minimizing property of u_n yields

$$\|Au_n - g_n\|^p + \mu\mathcal{R}(u_n) \le \left\|Au_\mu^\delta - g_n\right\|^p + \mu\mathcal{R}\left(u_\mu^\delta\right).$$

Hence the sequences $(\|Au_n - g_n\|^p)_n$ and $(\mathcal{R}(u_n))_n$ are uniformly bounded. Further, in view of the triangle inequality

$$\left\|Au_n - g^\delta\right\|^p \le 2^{p-1}\left(\|Au_n - g_n\|^p + \left\|g_n - g^\delta\right\|^p\right),$$

the sequence $\left(\left\|Au_n - g^\delta\right\|\right)_n$ is also uniformly bounded. Then, by the coercivity of the functional $\mathcal{J}_\mu$, the sequence $(u_n)_n$ is uniformly bounded in E, and there exists a subsequence, also denoted by $(u_n)_n$, converging weakly in E to some $u^* \in \mathcal{K}$. By the weak continuity of A and the convergence of g_n to g^δ in F, we have $Au_n - g_n \to Au^* - g^\delta$ weakly. In addition, the weak lower semicontinuity of norms and the operator $\mathcal{R}$ imply that

$$\begin{aligned}
\left\|Au^* - g^\delta\right\|^p + \mu\mathcal{R}(u^*) &\le \liminf_{n\to\infty} \|Au_n - g_n\|^p + \mu \liminf_{n\to\infty} \mathcal{R}(u_n) \\
&\le \liminf_{n\to\infty} \left(\|Au_n - g_n\|^p + \mu\mathcal{R}(u_n)\right) \\
&\le \liminf_{n\to\infty} \left\|Au_\mu^\delta - g_n\right\|^p + \mu\mathcal{R}\left(u_\mu^\delta\right) \\
&\le \left\|Au_\mu^\delta - g^\delta\right\|^p + \mu\mathcal{R}\left(u_\mu^\delta\right). \qquad (3.8)
\end{aligned}$$

Hence, the limit u^* is a minimizer to the functional $\mathcal{J}_\mu$. Suppose that the minimizer u_μ^δ is unique, then $u^* = u_\mu^\delta$, and every subsequence contains a subsubsequence that converges to u^* weakly. Therefore, the whole sequence converges weakly to the minimizer u_μ^δ.

Now under the assumption that $\mathcal{R}$ has the H-property, the proof is completed by showing that the functional value $(\mathcal{R}(u_n))_n$ converges to $\mathcal{R}(u^*)$. Conversely, suppose that

$$\limsup_{n\to\infty} \mathcal{R}(u_n) > \liminf_{n\to\infty} \mathcal{R}(u_n) \ge \mathcal{R}(u^*).$$

Then, there exists a subsequence $\left(u_{n_k}\right)$ of (u_n) such that

$$c := \limsup_{k\to\infty} \mathcal{R}\left(u_{n_k}\right) > \mathcal{R}(u^*).$$

Meanwhile, by taking $u_\mu^\delta = u^*$ in (3.8) yields

$$\lim_{k\to\infty} \left(\left\|Au_{n_k} - g_{n_k}\right\|^p + \mu\mathcal{R}\left(u_{n_k}\right)\right) = \left\|Au^* - g^\delta\right\|^p + \mu\mathcal{R}(u^*).$$

This two identities together yield

$$\lim_{k\to\infty} \left\|Au_{n_k} - g_{n_k}\right\|^p = \left\|Au^* - g^\delta\right\|^p + \mu\left(\mathcal{R}(u^*) - c\right) < \left\|Au^* - g^\delta\right\|^p,$$

which is in contradiction with the weak lower semicontinuity of norms, i.e.,

$$\left\|Au^* - g^\delta\right\| \le \liminf_{k\to\infty} \left\|Au_{n_k} - g_{n_k}\right\|.$$

As a result, we have $\lim_{n\to\infty} \mathcal{R}(u_n) = \mathcal{R}(u^*)$, which ends the proof. □

3.2.3 Consistency of a minimum

Now, we turn to the behavior of the minimizer u_μ^δ as the perturbation level δ goes to zero. A fundamental analysis on the variational regularization method is whether the approximate solution u_μ^δ converges to the true solution as δ tends to zero. This is referred to as the consistency in the literature.

Definition 3.2.2. *An element $u^\dagger \in E$ is called a $\mathcal{R}$-minimizing solution to the problem $Au = g^\dagger$ if it satisfies*

$$Au^\dagger = g^\dagger \quad \text{and} \quad \mathcal{R}\left(u^\dagger\right) \le \mathcal{R}(u) \quad \forall u \in \left\{u \in \mathcal{K} \mid Au = g^\dagger\right\}$$

Theorem 3.2.3. *Let the three assumption in Theorem 3.2.1 hold. Then, there exists at least one $\mathcal{R}$-minimizing solution to the problem $Au = g$.*

Proof. The existence of a $\mathcal{R}$ minimizing solution follows from theorem 3.2.1. Suppose, contrary to our claim, that there does not exist a $\mathcal{R}$-minimizing solution in $\mathcal{K}$. Then, there exists a sequence $(u_n)_n \subset \mathcal{K}$ of solutions to $Au = g^\dagger$ such that

$$\mathcal{R}(u_n)_n \to c \text{ and } c < \mathcal{R}(u), \quad \forall u \in \left\{u \in \mathcal{K} : Au = g^\dagger\right\}.$$

Hence, the functional $\mathcal{J}_\mu(u_n)$, for any fixed μ, with $g^\dagger$ in place of g^δ is uniformly bounded, and by the coercivity of $\mathcal{J}_\mu$, the sequence $(u_n)_n$ contains a subsequence,

also denoted by $(u_n)_n$, and some $u^* \in \mathcal{K}$ such that $u_n \to u^*$ weakly. Then, we have $Au^* = g^\dagger$ and $\mathcal{R}(u^*) = \lim_{n\to\infty} \mathcal{R}(u_n) = c$. This contradicts the assumption of the weak lower semicontinuity of $\mathcal{R}$ and completes the proof. □

Theorem 3.2.4. *Under the assumptions of Theorem 3.2.1, let $\left(g^{\delta_n}\right)_n \subset F$ be a sequence of disturbed data such that $\delta_n = \left\|g^\dagger - g^{\delta_n}\right\| \to 0$. Then the sequence of minimizers $\left(u_{\mu_n}^{\delta_n}\right)$ has a subsequence converging weakly to a $\mathcal{R}$-minimizing solution $u^\dagger$, if the regularization parameter $\mu_n \equiv \mu(\delta_n)$ satisfies*

$$\lim_{n\to\infty} \frac{\delta_n^p}{\mu_n} = 0 \quad \text{and} \quad \lim_{n\to\infty} \mu_n = 0.$$

Further, if the $\mathcal{R}$-minimizing solution $u^\dagger$ is unique, then the whole sequence converges weakly. Moreover, if the functional $\mathcal{R}$ satisfies the H-property, then the convergence is strong.

Proof. The minimizing property of the minimizer $u_{\mu_n}^{\delta_n}$ leads to

$$\begin{aligned}\left\|Au_{\mu_n}^{\delta_n} - g^{\delta_n}\right\|^p + \mu_n \mathcal{R}\left(u_{\mu_n}^{\delta_n}\right) &\le \left\|Au^\dagger - g^{\delta_n}\right\|^p + \mu_n \mathcal{R}\left(u^\dagger\right) \\ &\le \delta_n^p + \mu_n \mathcal{R}\left(u^\dagger\right).\end{aligned}$$

According to the definition of the sequence $(\mu_n)_n$, the sequences $\left(\left\|Au_{\mu_n}^{\delta_n} - g^{\delta_n}\right\|^p\right)_n$ and $\left(\mathcal{R}\left(u_{\mu_n}^{\delta_n}\right)\right)_n$ are both uniformly bounded. Further, we have

$$\left\|Au_{\mu_n}^{\delta_n} - g^\dagger\right\|^p \le 2^{p-1}\left(\left\|Au_{\mu_n}^{\delta_n} - g^{\delta_n}\right\|^p + \left\|g^{\delta_n} - g^\dagger\right\|^p\right).$$

Thus the sequence $\left(\left\|Au_{\mu_n}^{\delta_n} - g^\dagger\right\|^p\right)_n$ is also uniformly bounded. By the coercivity of the cost function, the sequence $\left(u_{\mu_n}^{\delta_n}\right)_n$ is uniformly bounded, and thus contains a subsequence, also denoted by $\left(u_{\mu_n}^{\delta_n}\right)_n$, converging weakly to some $u^* \in \mathcal{K}$ according to the convexity and closedness of the set $\mathcal{K}$. By the weak lower semicontinuity of norms, we deduce

$$\begin{aligned}\left\|Au^* - g^\dagger\right\|^p &\le \liminf_{n\to\infty} \left\|Au_{\mu_n}^{\delta_n} - g^{\delta_n}\right\|^p \\ &\le \limsup_{n\to\infty} \left\|Au_{\mu_n}^{\delta_n} - g^{\delta_n}\right\|^p + \mu_n \mathcal{R}\left(u_{\mu_n}^{\delta_n}\right) = 0.\end{aligned}$$

which means that $Au^* = g^\dagger$. It follows from the minimizing property of $u_{\mu_n}^{\delta_n}$ and weak lower semicontinuity of the functional $\mathcal{R}$, that

$$\mathcal{R}(u^*) \leq \liminf_{n\to\infty} \mathcal{R}\left(u_{\mu_n}^{\delta_n}\right) \leq \lim_{n\to\infty} \left(\frac{\delta_n^p}{\mu_n} + \mathcal{R}\left(u^\dagger\right)\right) = \mathcal{R}\left(u^\dagger\right). \tag{3.9}$$

Then u^* is a $\mathcal{R}$-minimizing solution. Now if the $\mathcal{R}$-minimizing solution $u^\dagger$ is unique, then $u^* = u^\dagger$. Clearly, every subsequence of $\left(u_{\mu_n}^{\delta_n}\right)$ contains a subsubsequence converging weakly to $u^\dagger$, and thus the whole sequence converges weakly. On the other hand, by taking $u^\dagger = u^*$ in the inequality (3.9), we obtain

$$\lim_{n\to\infty} \mathcal{R}\left(u_{\mu_n}^{\delta_n}\right) = \mathcal{R}(u^*).$$

This together with the H-property of $\mathcal{R}$ yield the strong convergence. □

3.3 Major examples of variational regularization problem

In the remainder of this section, different choices of the regularization term are presented.

3.3.1 Tikhonov regularization method

The general Tikhonov regularization problem takes the form of a penalized least squares minimization problem as follows

$$\min_u \left(\|Au - g^\delta\|_2^2 + \mu\|R(u)\|_2^2\right), \tag{3.10}$$

where R is a regularization linear operator. The interpretation of this minimization problem is that we are looking for a regularized solution that balances the significance of the two terms. The first term is given by the square of the residual norm that measures the goodness-of-fit of the solution u. On the one hand, if the residual is too large, then Au does not fit the data g^δ very well. On the other hand, if the residual is too small, then it is very likely that u is influenced by the error (perturbation) in the data. Otherwise, the second term involves a smoothing norm; the L^2 norm, and the regularization linear operator R. In general, R can be chosen as an approximation to a derivative operator[16].

Since the cost function of the minimization problem in (3.10) is linear and differentiable, the Euler-Lagrange formula can be used to compute the associated

normal equation

$$A^*A(u) + \mu R^*R(u) = A^*g^\delta. \tag{3.11}$$

3.3.2 Sparse regularization method

Another example of regularization method that has attracted a revived interest and considerable amount of attention in the literature, is ℓ^1 sparsity regularization. The sparse property of the solution is frequently used in the field of signal processing when it is known a priori that the desired solution is sparse or almost sparse in the following sense.

Definition 3.3.1. *[48] Let $(e_k)_{k\in\mathbb{N}}$ be a Schauder basis in E and let $(u_k)_{k\in\mathbb{N}}$ be the corresponding coefficients of some u in E. The Banach space element u is sparse with respect to the basis $(e_k)_{k\in\mathbb{N}}$ if only a finite number of coefficients do not vanish.*

According to Definition 3.3.1, sparsity is a property of the coefficient sequence with respect to a basis and not of a Banach space element itself. Thus, we may replace E with some sequence space such as the Banach space of absolutely summable real sequences ℓ_1 that known to be the best choice in the literature [48, 80]. One way to obtain such sparse approximations is by solving the following minimization problem

$$\min_{u\in\mathcal{K}}\left(\left\|Au - g^\delta\right\|_2^2 + \mu\|u\|_{\ell^1}\right), \tag{3.12}$$

where $\|u\|_{\ell^1} := \sum_k |u_k|$ stands for the ℓ^1-norm which is weakly lower semi-continuous. Then, the existence and stability of the minimizer follow from the verified assumptions in the previous theorems.

3.3.3 Total variation regularization method

The total variation regularizer was first introduced by Rudin, Osher, and Fatemi (ROF) in their well-known work [133] in 1992. It was designed with the explicit goal of preserving sharp discontinuities (edges) while removing noise and other unwanted fine scale detail. In other words, the total variation regularization term in the minimization discourages the solution from having oscillations, and it does allow the solution to have discontinuities. The regularization problem for total variation is given by

$$\min_{u\in\mathcal{K}}\left(\mathcal{J}_\mu(u) := \left\|Au - g^\delta\right\|_{L^1(\Omega)} + \mu|u|_{\mathrm{TV}(\Omega)}\right), \tag{3.13}$$

where $\Omega \subset \mathbb{R}^N$ is an open bounded domain, the space $\mathrm{BV}(\Omega)$ denotes the space of functions of bounded variation which is defined as follows

$$\mathrm{BV}(\Omega) = \left\{ v \in L^1(\Omega) \mid \|v\|_{L^1(\Omega)} + |v|_{\mathrm{TV}(\Omega)} < \infty \right\}, \tag{3.14}$$

where the total variation semi-norm $|\cdot|_{\mathrm{TV}(\Omega)}$ is given by

$$|v|_{\mathrm{TV}(\Omega)} = \sup_{\substack{w \in C_0^1(\Omega;\mathbb{R}^d) \\ \|w\|_{L^\infty(\Omega)} \leq 1}} \int v \, \mathrm{div}(\phi) d\lambda. \tag{3.15}$$

Theorem 3.3.1. *Let the operator A be a continuous operator with respect to $L^p(\Omega), 1 \leq p < \frac{N}{N-1}$ and $1 \notin \ker(A)$. Then the regularization problem (3.13) has at least one minimum.*

Proof. On account to Theorem 3.2.1, let us consider the space $E = \mathrm{BV}(\Omega)$ and $F = L^1(\Omega)$. By the compact embedding of the space $\mathrm{BV}(\Omega)$ into $L^p(\Omega)$, $1 \leq p < \frac{N}{N-1}$ [2], and the continuity of A in $L^p(\Omega)$, the operator A is weakly continuous. The assumption $1 \notin \ker(A)$ implies that functional is coercive on the space $\mathrm{BV}(\Omega)$. Furthermore, the regularization functional $|u|_{\mathrm{TV}}$ is BV weak$*$ lower semi continuous. Therefore, by Theorem 3.2.1, there exists at least one minimum of $\mathcal{J}_\mu$. □

The L^1 norm is usually avoided since the variation of expressions like $\int_\Omega |u| dx$ produces singular distributions as coefficients (e.g. δ distribution) which cannot be handled in a purely algebraic framework. However, the use of the BV semi-norm allows the recovery of the edges which is impossible if the second term in (3.13) were replaced by the strict convex and differentiable operator $R_p(u) = \int_\Omega |\nabla u|^p$ for any $p > 1$. The main reason for this effect is that for $p > 1$ the derivative of R_p corresponds to a nondegenerate elliptic differential operator of second order and thus has a smoothing effect in the optimality condition, whereas for total variation the operator is degenerate and affects only the level lines of the image [123].

The non differentiability issue of the Total Variation (TV) term is remedied in some works [62] by introducing a smoothing parameter in the TV term so that the cost functional becomes differentiable. The smoothed TV regularization term promotes the same regularization effects as the original one, the sparsity of the gradient of the solution as well as the well-defined edges. As a consequence, this new consideration allows the resolution of the minimization problem by classical gradient-based optimization methods.

Many algorithms for total variation image reconstruction have been developed, especially for the Gaussian noise model. To name a few, we found the effi-

cient primal-dual hybrid gradient algorithm for total variation image restoration [157], the split bregman algorithm, by Getreuer[54], and Chambolle's projection algorithm [28].

3.4 The regularization parameter selection

Besides a good choice of the regularization operator $\mathcal{R}$, the efficiency of a regularization method also strongly depends on an accurate selection of the regularization parameter, as we have shown before. Although different authors have proposed several approaches for proper regularization parameter selection. For instance, the Morozov discrepancy principle [119], Generalized cross validation (GCV) method [60], the L-curve method [69, 100], and Unbiased predictive risk estimation (UPRE) [109, 110], have been developed to automatically select an optimal regularization parameter. The drawback of most of them is that they can only be derived when the regularization term has a quadratic form.

3.4.1 Morozov's discrepancy principle

Morozov's discrepancy principle technique [119] selects the parameter μ by matching the norm of the residual to some upper bound, i.e., a good regularized solution u_μ^δ should verifies

$$\|Au_\mu^\delta - g^\delta\|_2^2 = c(\delta), \tag{3.16}$$

where c is a constant that depends on the noise level δ. When the variance σ^2 of the noise is available, the upper bound is given by $c(\delta) = \tau n_1 n_2 \sigma^2$, with τ being a predetermined parameter, one sets $\tau = 1$ [92]. If the variance σ^2 of the noise is unknown, it can be estimated using the median rule [116]. The existence of a solution $\mu(\delta)$ to equation (3.16) is given by the following result.

Theorem 3.4.1. *[79] If the function $\left\|Au_\mu^\delta - g^\delta\right\|$ is continuous with respect to μ, and $\lim_{\mu\to+\infty}\left\|Au_\mu^\delta - g^\delta\right\| > c(\delta)$ and $\lim_{\mu\to 0^+}\left\|Au_\mu^\delta - g^\delta\right\| < c(\delta)$, then there exists at least one positive solution $\mu(\delta)$ to equation (3.16).*

The equation (3.16) can be solved numerically using Newton's method [41, 151]. Let us consider the functional

$$\varphi(\mu) := \left\|Au_\mu^\delta - g^\delta\right\|^2 - c(\delta). \tag{3.17}$$

Using Newton's method, we have, for $j = 0, 1, 2, \ldots$

$$\mu_{j+1} = \mu_j - \frac{\varphi\left(\mu_j\right)}{\varphi'\left(\mu_j\right)}.$$

From another point of view, the regularization parameter can be seen as a Lagrange multiplier. Under the discrepancy principle, the variational regularization problem can be represented as solving a constrained optimization problem described as

$$\min_{u \in D} \mu \mathcal{R}(u), \tag{3.18}$$

where the subset $D = \left\{ u \in \mathcal{K},\ \left\| Au - g^{\delta} \right\|^2 \leqslant c(\delta) \right\}$.

3.4.2 Generalized cross validation

Another popular method for choosing the regularization parameter that does not require knowledge of the noise properties is the generalized cross-validation (GCV) [60]. The basic idea behind cross validation is to minimize the set of prediction errors that is, to choose μ so that the regularized solution obtained with a data point removed predicts this missing point well when averaged over all ways of removing a point. This viewpoint leads to the minimization with respect to μ of the following GCV function

$$\mathcal{G}(\mu) = \frac{\|g^{\delta} - A u_{\mu}^{\delta}\|_2^2}{\left[\operatorname{trace}\left(I - A\hat{A}\right)\right]^2}, \tag{3.19}$$

where $\hat{A}$ denotes the linear operator that generate the regularized solution when applied to data, so that $u_{\mu} = \hat{A} g^{\delta}$.

Remark 3.4.1. *Note that only the data are used in the calculation of $\mathcal{G}(\mu)$ and no prior knowledge of the noise amplitude, is required. However, there are also a number of difficulties related to the computation of the GCV cost in the equation (3.19). First, the operator $A\hat{A}$ must be found, while specifying this quantity is straightforward. In addition, it is well known that the GCV method tends to undersmooth the solution. Finally, in some cases the GCV cost curve is quite flat, leading to numerical problems in finding the minimum of $\mathcal{G}$, which can result in overly small values of μ. Sometimes, the GCV function can have multiple minimizers [94].*

3.4.3 L-curve method

Since all regularization methods involve a trade-off between fidelity to the data, as measured by the residual norm, and the fidelity to some prior information, as measured by the side constraint norm, it would seem natural to choose a regularization parameter based on the behavior of these two terms as μ is varied. Indeed, a graphical plot of $\|\mathcal{R}(u_\mu^\delta)\|_2$ versus $\|g^\delta - Au_\mu^\delta\|_2$ on a log−log scale as μ is varied is called the L-curve and has been proposed as a means to choose the regularization parameter [71]. The L-curve, shown schematically in Figure

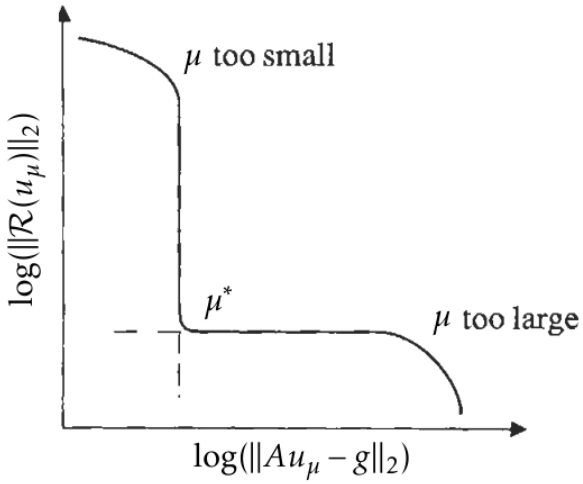


Figure 3.1: The generic form of the L-curve.

3.1, has a characteristic "L" shape (hence its name), which consists of a vertical part and a horizontal part. The vertical part corresponds to under-regularized estimates, where the solution is dominated by the amplified noise. In this region, small changes to μ have a large effect on the size or energy of u, but a relatively small impact on the data fit. The horizontal part of the L-curve corresponds to over smoothed estimates, where the solution is dominated by residual fit errors. In this region changes to μ affect the size of $\hat{u}$ weakly, but produce a large change in the fit error.

The idea behind the L-curve approach for choosing the regularization parameter is that the corner between the horizontal and vertical portions of the curve defines the transition between over and under regularization, and thus represents a balance between these two extremes and the best choice of μ. The point on the curve corresponding to this μ is shown as μ^* in Figure 3.1. While the notion of choosing μ to correspond to the corner of the L-curve is natural and intuitive, there exists the issue of defining exactly what is meant by the "corner" of this curve. A number of definitions have been proposed, including the point

of maximum curvature, the point closest to a reference location, and the point of tangency , etc. The last definition is especially interesting, since it can be shown that the optimal μ for this criterion must satisfy

$$\mu^2 = \frac{\|Au_\mu - g\|_2^2}{\|\mathcal{R}(u_\mu)\|_2^2}. \tag{3.20}$$

The main difficulty with the L-curve method is that we need to solve (3.1) many times for different μ values, and therefore, the algorithm is computationally expensive. Sometimes, it is difficult to locate the corner or there does not exist a corner. In addition, the regularized solutions obtained by the L-curve approach fail to converge to the exact solution when the noise level $\delta \to 0$ (see [45, 68] for more details).

Remark 3.4.2. *In the numerical point of view, and essentially in the case of iterative resolution processes, it is useful to consider a sequence of regularization parameter instead of fixed value. This consideration will improve the convergence in practice, as well as the performance will be less dependent on the initial choice.*

CHAPTER 4

High order tensor algebra

Among the most interesting points that have fueled interest in tensor representation and analysis are the increased ability of data collection systems to store large volumes of multidimensional data, as well as the accurate modeling that can be provided by leaving the data in its natural; multidimensional form. In the remainder of this chapter, we recall some interesting preliminaries and notation of tensor algebra [35, 97, 103].

4.1 Definitions and basic properties

Definition 4.1.1 (Tensor). *Let $I_1, I_2, \ldots, I_N \in \mathbb{N}$, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an Nth-order tensor of size $I_1 \times I_2 \times \cdots \times I_N$, its entries are denoted $\mathcal{X}_{i_1,\ldots,i_N}$ or $\mathcal{X}(i_1,\ldots,i_N)$ with $i_n \in (1,\ldots,I_n)$ for every $1 \leq n \leq N$.*

In other words, an Nth-order tensor is multidimensional array of size $I_1 \times I_2 \times \cdots \times I_N$. For example, a scalar is a tensor of zero-order, a vector is a tensor of first-order, a matrix is a second-order tensor, and a tensor of order three or higher are called high-order tensor.

Definition 4.1.2 (Order, Mode, Fibers and Slices). *Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an Nth-order tensor.*

- *The **order** of a tensor is the number of dimensions. For example, the order of $\mathcal{X}$ equal to N.*
- *Each dimension (way) is called a **mode**.*
- ***Fibers** are the higher-order analog of matrix rows and columns, it is defined by fixing every index but one.*

- ***Slices*** *are two-dimensional section of a tensor, it is defined by fixing all but two indices.*

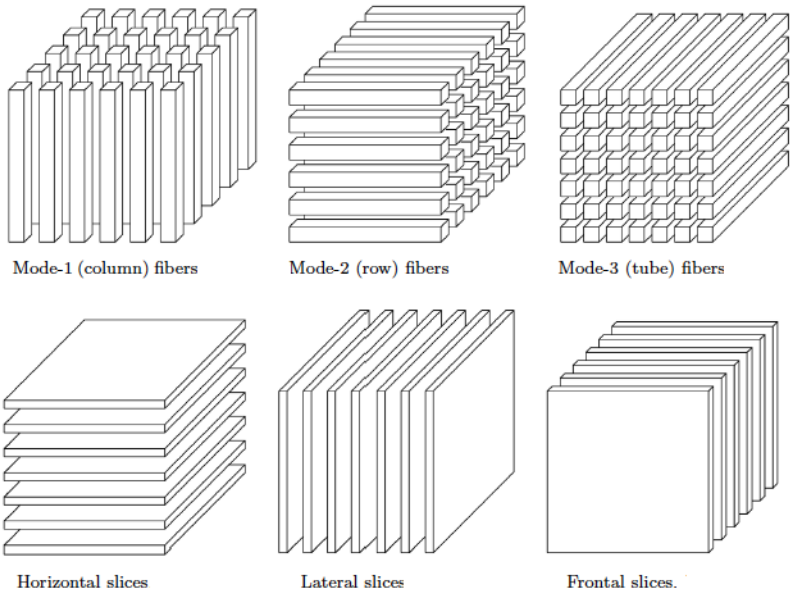


Figure 4.1: Fibers and slices of a third-order tensor.

Definition 4.1.3 (Symmetric tensor)**.** *A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is symmetric if its entries are invariant over any permutation of its indices, i.e, for every permutation p, $\mathcal{X}$ satisfied :*

$$\mathcal{X}_{i_1,\ldots,i_N} = \mathcal{X}_{i_{p(1)},\ldots,i_{p(N)}} \text{ for all } i_1,\ldots,i_N \in \{1,\ldots,N\},$$

with $p \in \pi_N$ where π_N the symmetric group of degree N, and cardinal $N!$.

Definition 4.1.4 (Diagonal tensor)**.** *A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is diagonal if $\mathcal{X}_{i_1 \ldots i_N} \neq 0$ for $i_1 = \cdots = i_N$ and zeros elsewhere. We use $\mathcal{I}$ to denote the identity tensor with ones on the superdiagonal and zeros elsewhere.*

A tensor $\mathcal{X}$ is called cubical if every mode has the same size, i.e. $I_1 = \cdots = I_N = I$, where $\mathcal{X} \in \mathbb{R}^{I \times I \times \cdots \times I}$.

Definition 4.1.5. *Let $k \geq 1$ be an integer. The norm of the tensor $\mathcal{X}$ is defined by*

$$|||\mathcal{X}|||_k = \left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} |\mathcal{X}(i_1, i_2, \ldots, i_N)|^k \right)^{1/k},$$

where $|\cdot|$ *stands for the absolute value in* $\mathbb{R}$. *The* ***inner product*** *of two same sized tensors* $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *is defined by*

$$\langle \mathcal{X} | \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{X}(i_1, i_2, \ldots, i_N) \mathcal{Y}(i_1, i_2, \ldots, i_N).$$

It follows immediately that $|||\mathcal{X}|||_2 = \sqrt{\langle \mathcal{X} | \mathcal{X} \rangle}$.

For simplicity of notation, we let $\|\,.\,\|_F$ stand for the tensor norm in the case when $k = 2$, namely, $\|\,.\,\|_F = |||\,.\,|||_2$. On the other hand, in the case when $N = 2$, which means 2 order tensors (matrices), it is important to make it clear that the norm $|||\,.\,|||_2$ is exactly the well-known Frobenius norm but not the induced matrix norm $\|\,.\,\|_2$.

Definition 4.1.6 (The mode-n matrix (Matricization))**.** *The mode-n matricization (also known as mode-n unfolding) of a tensor* $\mathcal{X}$ *is denoted by the matrix* $\mathcal{X}_{(n)}$ *of size* $I_n \times I_1 \cdot I_2 \cdots I_{n-1} \cdot I_{n+1} \cdots I_N$, *and it is defined by*

$$\mathcal{X}_{(n)}(i_n, j) = \mathcal{X}(i_1, i_2, \ldots, i_n, \ldots, i_N) \text{ where } j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m.$$

In other words, the mode-n matricization can be defined as the operation of arranging the mode-n fibers to be the columns of the matrix $\mathcal{X}_{(n)}$.

Example 4.1.1. *The mode-n matricization of a third-order tensor* $\mathcal{X}$ *of size* $I_1 \times I_2 \times I_3$ *can be illustrated as in Figure 4.2. Thus, we can express mode-n matrices using slices:*

$$\begin{aligned}
\mathcal{X}_{(1)} &= [\mathcal{X}(:,:,1), \mathcal{X}(:,:,2), \ldots, \mathcal{X}(:,:,I_3)], \\
\mathcal{X}_{(2)} &= \left[\mathcal{X}(:,:,1)^T, \mathcal{X}(:,:,2)^T, \ldots, \mathcal{X}(:,:,I_3)^T\right], \\
\mathcal{X}_{(3)} &= [\mathcal{X}(1,:,:), \mathcal{X}(2,:,:), \ldots, \mathcal{X}(I_1,:,:)].
\end{aligned}$$

Similar to the vectorisation of a matrix, the vectorisation of a tensor is a linear transformation that converts the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into a column vector. The vectorization of $\mathcal{X}$ is denoted by $\boldsymbol{vec}(\mathcal{X})$, where

$$\boldsymbol{vec} : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}^{I_1 \cdot I_2 \cdots I_N}$$

stacks the entries of a tensor in reverse lexicographical order into a long column vector.

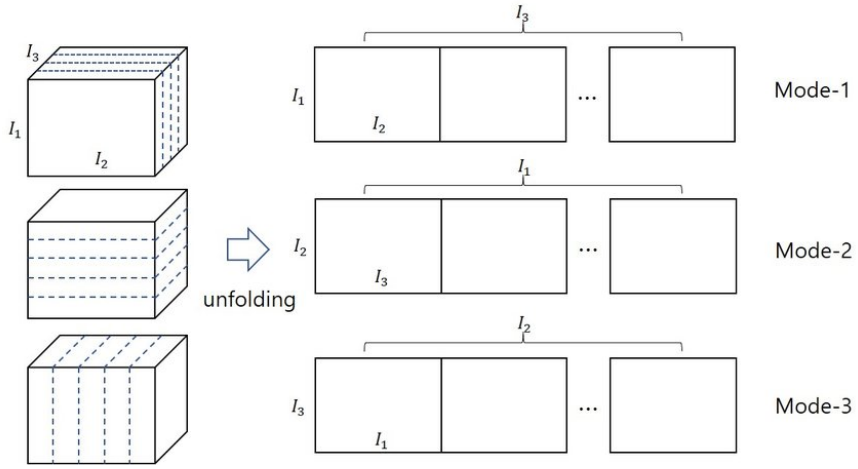


Figure 4.2: Illustration of matricizing a third-order tensor into a matrix in three modes.

4.2 Tensor products and multiplications

The question that arises when one seeks to generalize an equation or a model in a higher dimension is the operation that must be used between these multidimensional data. In the following section, we will introduce a class of different operations that will be useful for generalizing our model in high dimensions.

Definition 4.2.1 (The outer product)**.** *The outer product of a tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *and* $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_M}$ *is a tensor denoted by* $\mathcal{Z} = \mathcal{X} \circ \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times J_1 \times J_2 \times \cdots \times J_M}$. *Elementwise,*

$$\mathcal{Z}(i_1, i_2, \ldots, i_N, j_1, j_2, \ldots, j_M) = \mathcal{X}(i_1, i_2, \ldots, i_N)\mathcal{Y}(j_1, j_2, \ldots, j_M).$$

In particular, let $u \in \mathbb{R}^I$ and $v \in \mathbb{R}^J$ be two column vectors, their outer product is given by the matrix $W = u \circ v \in \mathbb{R}^{I \times J}$ defined by $W = uv^T$, namely,

$$(u \circ v)_{i,j} = u(i)v(j).$$

Definition 4.2.2. *An Nth-order tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *is rank one if it can be written as the outer product of N vectors, which means* $\mathcal{X} = x_1 \circ x_2 \circ \cdots \circ x_N$, *where*

$$\mathcal{X}(i_1, i_2, \ldots, i_N) = \prod_{k=1}^{N} x_k(i_k) \text{ for all } 1 \leq i_k \leq I_N, \tag{4.1}$$

and $x_k(i_k)$ *denotes the* i_k^{th} *element of vector* $x_k \in \mathbb{R}^{I_k}$.

Definition 4.2.3 (The contracted product)**.** *The contracted product of two tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_M}$ with $I_N = J_1$, is denoted by the tensor $\mathcal{Z} = \mathcal{X} \bullet \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N-1} \times J_2 \times \cdots \times I_M}$. Elementwise, it is defined by*

$$\mathcal{Z}(i_1, i_2, \ldots, i_{N-1}, j_2, \ldots, j_M) = \sum_{i_k=1}^{I_N} \mathcal{X}(i_1, i_2, \ldots, i_{N-1}, i_k)\mathcal{Y}(i_k, j_2, \ldots, j_M),$$

for $i_n = 1, 2, \ldots, I_N$, $j_m = 1, 2, \ldots, I_M$ and $m = 2, \ldots, M$.

Notice that the contracted product is a generalization of the usual matrix product.

Definition 4.2.4 (Kronecker product)**.** *Let $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ be two matrices. Their Kronecker product is a matrix of size $mp \times nq$ given by*

$$A \otimes B = \begin{pmatrix} a_{11}B & \ldots & a_{1n}B \\ a_{21}B & \ldots & a_{2n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \ldots & a_{mn}B \end{pmatrix}$$

Definition 4.2.5 (Khatri-Rao product)**.** *Let $A = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^{m \times n}$ and $B = [b_1, b_2, \ldots, .b_n] \in \mathbb{R}^{p \times n}$ be two matrices, their Khatri-Rao product is a matrix of size $mp \times n$ defined by*

$$A \odot B = [a_1 \otimes b_1, a_2 \otimes b_2, \ldots, a_n \otimes b_n].$$

Remark 4.2.1. *If a and b are vectors, then the Khatri-Rao product and the Kronecker product are identical, i.e, $a \otimes b = a \odot b$.*

Definition 4.2.6 (Hadamard product)**.** *Given matrices A and B, both of size $I \times J$, their Hadamard product is denoted by the matrix $A \circledast B$ also of size $I \times J$ and defined by:*

$$A \circledast B = \begin{pmatrix} a_{11}b_{11} & \ldots & a_{1n}b_{1n} \\ a_{21}b_{21} & \ldots & a_{2n}b_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & \ldots & a_{mn}b_{mn} \end{pmatrix}$$

Proposition 4.2.1. *Let A,B,C be a three matrices, we have the following properties*

1. $vec(ABC) = (C^T \otimes A)vec(B)$
2. $(A \otimes B) \otimes C = A \otimes (B \otimes C)$.
3. $(A \odot B) \odot C = A \odot (B \odot C)$.

4. $(A \otimes B)^T = A^T \otimes B^T$.

5. $(A \odot B)^T (A \odot B) = (A^T A) \circledast (B^T B)$.

6. $(A \odot B)^\dagger = ((A^T A) \circledast (B^T B))^\dagger (A \odot B)^T$ *with* $A^\dagger$ *is the pseudo-inverse of* A.

Proof. Readers can refer to [77] for more details about the proof. □

4.2.1 The n-mode product

Definition 4.2.7 (The n-mode product matrix)**.** *Let* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *be an* N*th-order tensor and* $A \in \mathbb{R}^{J \times I_n}$ *be a matrix. The n-mode product of the tensor* $\mathcal{X}$ *by the matrix* A *is an* N*th-order tensor, denoted by* $\mathcal{X} \times_n A$*, of size* $I_1 \times I_2 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ *and whose entries are given by*

$$(\mathcal{X} \times_n A)_{i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1,\ldots,i_N} A_{j i_n},$$

i.e, each mode-n fiber of $\mathcal{X}$ *is multiplied by the matrix* A.

Proposition 4.2.2. *[97, 102, 98] Given two tensors* $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$*, we have the following properties:*

1. $(\mathcal{X} \times_n A) \times_m B = (\mathcal{X} \times_m B) \times_n A$*, for all* $A \in \mathbb{R}^{J_n \times I_n}$ *and* $B \in \mathbb{R}^{J_m \times I_m}$*, with* $m \neq n$.

2. $(\mathcal{X} \times_n A) \times_n B = \mathcal{X} \times_n (BA)$*, for all* $A, B \in \mathbb{R}^{J_n \times I_n}$.

3. $\mathcal{X} \times_n I_d = \mathcal{X}$ *where* $I_d \in \mathbb{R}^{I_n \times I_n}$ *stands for the identity matrix.*

4. $\langle \mathcal{X} \times_n A, \mathcal{Y} \rangle = \langle \mathcal{X}, \mathcal{Y} \times_n A^T \rangle$*, for all* $A \in \mathbb{R}^{I_n \times I_n}$.

5. *If* $A \in \mathbb{R}^{I_n \times J_n}$ *is full column rank matrix, then*

$$\mathcal{X} = \mathcal{Y} \times_n A \Rightarrow \mathcal{Y} = \mathcal{X} \times_n A^\dagger.$$

6. *If the matrix* $A \in \mathbb{R}^{I \times J_n}$ *is orthonormal, then*

$$\mathcal{X} = \mathcal{Y} \times_n A \Rightarrow \mathcal{Y} = \mathcal{X} \times_n A^\top.$$

7. *If the matrix* A *is orthogonal, then,* $\|\mathcal{X} \times_n A\|_F^2 = \|\mathcal{X}\|_F^2$.

Proof. The proof is straightforward using the elementwise definition. □

The next property is among the most important proportions that link the tensor structure to the matrix structure and therefore facilitate the analysis and manipulation of the tensor equation.

Proposition 4.2.3. *Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and the sequence of matrices $A^{(n)} \in \mathbb{R}^{J_n \times I_n}$ for all $n \in \{1,\ldots,N\}$. Then, for any $n \in \{1,\ldots,N\}$, we have*

$$\mathcal{Y} = \mathcal{X} \times_1 A^{(1)} \times_1 \cdots \times_N A^{(N)} \Leftrightarrow \mathcal{Y}_{(n)} = A^{(n)} \mathcal{X}_{(n)} \left(A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)} \right)^{\top}.$$

Proof. For more details, we refer the reader to see [98]. □

Definition 4.2.8 (The n-mode product vector)**.** *Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an Nth-order tensor and $v \in \mathbb{R}^{I_n}$ be a vector. The $N-1$ order tensor $\mathcal{X} \overline{\times}_n v$ denotes the n-mode product of $\mathcal{X}$ by v, where it is a tensor of size $I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N$ whose entries are given by*

$$(\mathcal{X} \overline{\times}_n v)_{i_1,\ldots,i_{n-1},i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1,\ldots,i_N} v_{i_n}. \tag{4.2}$$

4.2.2 Tensor t-product

This product based on the discrete Fourier transform (*DFT*) in order to reduce the computation of the associated tensor operation. Let us first recall the definition and some properties of the discrete Fourier transform.

Definition 4.2.9. *The discrete Fourier transform matrix, called also the Fourier matrix, is defined by the m-by-m square matrix F_m with entries given by*

$$F_m(j,k) = e^{2\pi i jk/m}, \ \forall \ j,k \leqslant m, \tag{4.3}$$

where i is the imaginary number. The matrix F_m is normalized by $1/\sqrt{m}$, namely

$$\left(\frac{1}{\sqrt{m}} F_m\right)\left(\frac{1}{\sqrt{m}} F_m\right)^T = I_d.$$

Fast algorithms have been developed to reduce the complexity of computing the discrete Fourier transform from $O\left(m^2\right)$ to $O(m \log m)$. For example, for any m vector v, the matrix-vector multiplications $F_m v$, which is the discrete Fourier transform of the vector v, can be computed in $O(m \log m)$ operations by Cooley–Tukey Fast Fourier Transform (FFT) algorithm [53].

Definition 4.2.10. *[40] A circulant matrix define as a matrix in which every row is the same as the previous row, just shifted to the right by one. Let $w = (w_1,\ldots,w_m)^T$*

be an m-vector, the circulant operation on w defined by the m-by-m circulant matrix with w as the first column:

$$\boldsymbol{circ}(w) = \begin{pmatrix} w_1 & w_m & \cdots & w_2 \\ w_2 & w_1 & \cdots & w_3 \\ \vdots & & & \vdots \\ w_m & w_{m-2} & \cdots & w_1 \end{pmatrix}. \tag{4.4}$$

Let us denote by $\Gamma_m(\mathbb{K})$ *the space of m-by-m circulant matrix define as*

$$\Gamma_m(\mathbb{K}) = \{C \in \mathbb{K}^{m\times m};\ C = \boldsymbol{circ}(w),\ w \in \mathbb{K}^m\}. \tag{4.5}$$

Theorem 4.2.1. *[40] For any integer* $m \in \mathbb{N}$*, the vector space containing all matrices that can be diagonalized by* F_m*, is equal to* $\Gamma_m(\mathbb{K})$*, i.e.*

$$\Gamma_m(\mathbb{K}) = \{F_m \Lambda_m F_m^* \quad | \quad \Lambda_m \text{ is an } m\text{-by-}m \text{ diagonal matrix.}\} \tag{4.6}$$

The previous theorem proves that the Fourier matrix diagonalizes any circulant matrix, as a consequence, all matrices in $\Gamma_m(\mathbb{K})$ have the same unitary eigenvectors basis represent by F_m [40]. On the other side, the eigenvalues of each circulant matrix are the Discrete Fourier Transformation (DFT) of its first column, i.e. for any $w \in \mathbb{K}^m$, then the eigenvalues of the circulant matrix $Cr_m(w)$

$$\Lambda_m = \text{diag}(F_m Cr_m(w) e_1) = \text{diag}(F_m w), \tag{4.7}$$

which means that we can compute the eigenvalues of $\Gamma_m(w)$ without using all the elements of the matrix.

Based on the block circulant structure, the t-product for high order tensors defined as follows.

Definition 4.2.11. *[96] The t-product of two tensors* $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times I_3}$ *and* $\mathcal{B} \in \mathbb{R}^{I_2\times J\times I_3}$*, is a tensor* $\mathcal{A} *_t \mathcal{B} \in \mathbb{R}^{I_1\times J\times I_3}$ *defined as*

$$\mathcal{A} *_t \mathcal{B} = \boldsymbol{ibvec}\left(\boldsymbol{bcirc}(\mathcal{A}) \cdot \boldsymbol{bvec}(\mathcal{B})\right), \tag{4.8}$$

where the dot $\cdot$ *stands for the usual matrix product, the matrix* $\boldsymbol{bcirc}(\mathcal{A})$ *is a block circulant matrix defined by using the frontal slices of* $\mathcal{A}$ *and* $\boldsymbol{bvec}(\mathcal{B})$ *defined as a block vector contain the frontal slices of* $\mathcal{B}$*, such as:*

$$\boldsymbol{bcirc}(\mathcal{A}) = \begin{pmatrix} A_1 & A_{I_3} & \cdots & A_2 \\ A_2 & A_1 & \cdots & A_3 \\ \vdots & \ddots & & \vdots \\ A_{I_3} & A_{I_3-1} & \cdots & A_1 \end{pmatrix} \quad \textit{and} \quad \boldsymbol{bvec}(\mathcal{B}) = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_{I_3} \end{pmatrix}.$$

While ***ibvec*** *stand for the inverse operation of* ***bvec*** *that transform a* $\boldsymbol{bvec}(\mathcal{B})$ *into a tensor* $\mathcal{B}$.

Just as circulant matrices can be diagonalized by the DFT, block circulant matrices can be block diagonalized as follows

$$\left(F_{I_3} \otimes \mathbf{I}_{I_1}\right) \cdot \boldsymbol{bcirc}(\mathcal{A}) \cdot \left(F_{I_3}^* \otimes \mathbf{I}_{I_2}\right).$$

As a consequence, the product defined in (4.8), can be computed in $O(I_1 I_2 I_3 \log_2(I_3))$ flops using the *FFT*.

Definition 4.2.12. *[96] The conjugate transpose of a tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ *is the* $\mathcal{A}^* \in \mathbb{R}^{I_2 \times I_1 \times I_3}$ *obtained by conjugate transposing each of the frontal slice and then reversing the order of transposed frontal slices 2 through* I_3.

Definition 4.2.13. *The* $I \times I \times J$ *identity tensor* $\mathcal{I}$ *is the tensor whose the first frontal slice is the* $I \times I$ *identity matrix, and the other frontal slices are all zeros. It is clear that* $\mathcal{A} *_t \mathcal{I} = \mathcal{I} *_t \mathcal{A} = \mathcal{A}$ *given the appropriate dimensions.*

For an $I \times I \times J$ tensor, an inverse exists if it satisfies the following definition.

Definition 4.2.14. *An* $I \times I \times J$ *tensor* $\mathcal{A}$ *has an inverse* $\mathcal{B}$ *provided that* $\mathcal{A} *_t \mathcal{B} = \mathcal{I}$, *and* $\mathcal{B} *_t \mathcal{A} = \mathcal{I}$.

Proposition 4.2.4. *[96, 95] Let* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, $\mathcal{B} \in \mathbb{R}^{I_2 \times J \times I_3}$ *and* $\mathcal{C} \in \mathbb{R}^{J \times K \times I_3}$ *be third order tensors. We have*

- *The t-product is associative:* $(\mathcal{A} *_t \mathcal{B}) *_t \mathcal{C} = \mathcal{A} *_t (\mathcal{B} *_t \mathcal{C})$.
- *The set of all invertible* $I \times I \times J$ *tensors forms a group under* $*_t$ *operation.*
- *If* $\mathcal{A}$ *is an orthogonal tensor, then* $\|\mathcal{A} *_t \mathcal{B}\|_F = \|\mathcal{B}\|_F$.

We say a tensor is "f-diagonal" if each frontal slice is diagonal. Likewise, a tensor is f-upper triangular or f-lower triangular if each frontal slice is upper or lower triangular, respectively.

Theorem 4.2.2 (Tensor Singular Value Decomposition (t-SVD))**.** *[96] Let* $\mathcal{A}$ *be an* $I_1 \times I_2 \times I_3$ *real-valued tensor. Then* $\mathcal{A}$ *can be factored as*

$$\mathcal{A} = \mathcal{U} *_t \mathcal{S} *_t \mathcal{V}^T,$$

where $\mathcal{U}, \mathcal{V}$ *are orthogonal* $I_1 \times I_1 \times I_3$ *and* $I_2 \times I_2 \times I_3$, *respectively, and* $\mathcal{S}$ *is a* $I_1 \times I_2 \times I_3$ *f-diagonal tensor.*

Remark 4.2.2. *On the one hand, all the definition provided in this paragraph developed for the third-order tensor. However, all these notions can generalize to higher-order tensors in a recursive manner. On the other hand, based on the same concept, the authors in [95] developed the cosine-product using Discrete Cosine Transformation (DCT).*

4.2.3 Tensor Einstein product

Definition 4.2.15. *Let the tensors* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_L \times K_1 \times K_2 \times \ldots \times K_N}$ *and* $\mathcal{B} \in \mathbb{R}^{K_1 \times K_2 \times \ldots \times K_N \times J_1 \times J_2 \times \ldots \times J_M}$*, the Einstein product of tensors* $\mathcal{A}$ *and* $\mathcal{B}$ *is a tensor of size* $\mathbb{R}^{I_1 \times I_2 \times \ldots \times I_L \times J_1 \times J_2 \times \ldots \times J_M}$ *whose elements are defined by*

$$(\mathcal{A} *_N \mathcal{B})_{i_1 \ldots i_L j_1 \ldots j_M} = \sum_{k_1, \ldots, k_N} a_{i_1 \ldots i_L k_1 \ldots k_N} b_{k_1 \ldots k_N j_1 \ldots j_M}.$$

Definition 4.2.16. *Given a tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$*, the tensor* $\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_M \times I_1 \times \cdots \times I_N}$ *is the transpose of* $\mathcal{A}$*, if* $b_{i_1 \ldots i_M j_1 \ldots j_m} = a_{j_1 \ldots j_N i_1 \ldots i_M}$*. We denote the transpose of* $\mathcal{A}$ *by* $\mathcal{A}^T$*.*

Definition 4.2.17. *A tensor* $\mathcal{D} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_N}$ *is said to be diagonal if all of its entries are equal to zero except for* $d_{i_1 \ldots i_N i_1 \ldots i_N}$*. In the case* $d_{i_1 \ldots i_N i_1 \ldots i_N} = 1$*, the tensor* $\mathcal{D}$ *is called the identity tensor and denoted by* $\mathcal{I}_N$*. We further use the notation* $\mathcal{O}$ *for a the tensor having all its entries equal to zero.*

Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_1 \times I_2 \times \ldots \times I_N}$. The tensor $\mathcal{A}$ is invertible if there exists a tensor $\mathcal{X} \in \mathbb{R}_1 \times I_2 \times \ldots \times I_N \times I_1 \times I_2 \times \ldots \times I_N$ such that

$$\mathcal{A} *_N \mathcal{X} = \mathcal{X} *_N \mathcal{A} = \mathcal{I}_N.$$

4.3 Tensor decomposition

In addition to the t-SVD tensor decomposition, related to the t-product, recalled in the previous section, two particular tensor decompositions can be considered to be higher-order extensions of the matrix singular value decomposition: CANDECOMP/PARAFAC (CP) decomposes a tensor as a sum of rank-one tensors[35], and the Tucker decomposition is a higher-order form of principal component analysis [142]. We will discuss these two decompositions in the following.

4.3.1 Tucker decomposition

The Tucker decomposition decomposes a tensor into a core tensor multiplied by a matrix along each mode.

Definition 4.3.1. *Let $I_1, I_2, \ldots, I_N \in \mathbb{N}$, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an Nth-order tensor. The Tucker decomposition of $\mathcal{X}$ is given by*

$$\mathcal{X} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)} = [|\mathcal{G}; U^{(1)}, U^{(2)}, \ldots, U^{(N)}|], \tag{4.9}$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is called the core tensor it is contains the mode-n singular values of $\mathcal{X}$, $U^{(n)} \in \mathbb{R}^{I_n \times R_n}$ are factor matrices and $(R_1, ..., R_N)$ is the multi-linear rank of the Tucker decomposition.

The mode-n matricized version is given by

$$\mathcal{X}_{(n)} = U^{(n)} \mathcal{G}_{(n)} (U^{(N)} \otimes \cdots \otimes U^{(n+1)} \otimes U^{(n-1)} \otimes \cdots \otimes U^{(1)})^T.$$

Remark 4.3.1 (Non-uniqueness of Tucker decomposition)**.** *Let $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ be an third-order tensor. The Tucker decomposition of $\mathcal{X}$ is given by*

$$\mathcal{X} \approx \mathcal{G} \times_1 A \times_2 B \times_3 C = [|\mathcal{G}, A, B, C|]. \tag{4.10}$$

Let $U \in \mathbb{R}^{P \times P}$, $V \in \mathbb{R}^{Q \times Q}$ and $W \in \mathbb{R}^{R \times R}$ be nonsingular matrices. Then, we obtain

$$[|\mathcal{G}, A, B, C|] = [|\mathcal{G} \times_1 U \times_2 V \times_3 W; AU^{-1}, BV^{-1}, CW^{-1}|].$$

*In other words, we can easily construct two decomposition of $\mathcal{X}$ completely different. Therefore, the Tucker decompositions **are not unique** [35, 97].*

Definition 4.3.2 (Higher-Order SVD (HOSVD))**.** *Let $I_1, I_2, ..., I_N \in \mathbb{N}$, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times .... \times I_N}$ is an Nth-order tensor and $U^{(n)} \in \mathbb{R}^{I_n \times I_n}$ be a unitary matrix containing a basis of the left singular vectors of the frontal slices $X_{(n)}$. The core tensor is defined as*

$$\mathcal{G} = \mathcal{X} \times_1 {U^{(1)}}^T \times_2 {U^{(2)}}^T \ldots \times_N {U^{(N)}}^T \in \mathbb{R}^{I_1 \times I_2 \times .... \times I_N}.$$

Then, the HOSVD of $\mathcal{X}$ expressed by the following decomposition

$$\mathcal{X} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 \ldots \times_N U^{(N)}.$$

4.3.2 CP decomposition

CP decomposes a tensor into a sum of component rank-one tensors[35].

Definition 4.3.3. *Let $I_1, I_2, \ldots, I_N \in \mathbb{N}$, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an Nth-order tensor. The CP decomposition of $\mathcal{X}$ is given by*

$$\mathcal{X} = \sum_{r=1}^{R} a_r^1 \circ a_r^2 \circ \cdots \circ a_r^N, \text{ with } R \leq \prod_{n=1}^{N} I_n, \tag{4.11}$$

and a_r^k are vectors of size I_k for all $1 \leq k \leq N$. Elementwise, we have

$$\mathcal{X}_{i_1,\ldots,i_N} = \sum_{r=1}^{R} a_r^1(i_1)a_r^2(i_2)\cdots a_r^N(i_N), \text{ for all } i_1,\ldots,i_N.$$

If we define $A_n = [a_1^n, a_2^n, \ldots, a_R^n]$ for every $n \in \{1,\ldots,N\}$, the CP decomposition of $\mathcal{X}$ becomes

$$\mathcal{X} = A_1 \circ A_2 \circ \cdots \circ A_N,$$

where the matrices $A_n \in \mathbb{R}^{I_n \times R}$ is called the factor matrix. Then, we have:

$$\mathcal{X} \approx [|\lambda, A_1, A_2, \ldots, A_N|] \equiv \sum_{r=1}^{R} \lambda_r a_r^1 \circ a_r^2 \circ \cdots \circ a_r^N, \text{ where } \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_R] \in \mathbb{R}^R.$$

The mode-n unfolding version is given by

$$\mathcal{X}_{(n)} \approx A_n \Lambda (A_N \odot \cdots \odot A_{n+1} \odot A_{n-1} \odot \cdots \odot A_1)^T,$$

where $\Lambda = diag(\lambda)$.

Remark 4.3.2. *It is interesting to note that the CP decomposition can be viewed as a special case of Tucker, where the core tensor is superdiagonal and $R_1 = R_2 = \cdots = R_N = R$.*

Definition 4.3.4. *The rank of a tensor $\mathcal{X}$ denoted $rank(\mathcal{X})$, is the smallest number of rank-one tensors that generate $\mathcal{X}$ as their sum. In other words, the rank define as the smallest number of components in an CP decomposition.*

The definition of tensor rank is an exact analogue to the definition of matrix rank, but the properties of matrix and tensor ranks are quite different. One difference is that the rank of a real-valued tensor may actually be different over $\mathbb{R}$ and $\mathbb{C}$. Another major difference between matrix and tensor rank is that there is no simple algorithm to determine the rank of a specific given tensor. In fact, the problem is NP-hard which makes it impractical for most applications. Thus, we define the n-rank and the multi-linear rank.

Definition 4.3.5 (Tensor n-rank)**.** *The n-rank of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is defined as the number of linearly independent n-mode fibres of a tensor $\mathcal{X}$:*

$$rank_n(\mathcal{X}) = rank(\mathcal{X}_{(n)}), \text{ where } rank_n(\mathcal{X}) \leq \min(I_n, \prod_{\substack{i=1 \\ i \neq n}}^{N} I_i).$$

Definition 4.3.6 (Multi-linear rank)**.** *The multi-linear rank of a tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *is a vector noted* $rank_{\mathcal{X}}$ *and is given by:*

$$rank_{\mathcal{X}} = (rank(\mathcal{X}_{(1)}), rank(\mathcal{X}_{(2)}), ..., rank(\mathcal{X}_{(N)}))$$

Where $rank(\mathcal{X}_{(n)})$*, for* $1 \leq n \leq N$ *is the rank of the unfolding matrix* $\mathcal{X}_{(n)}$*.*

Remark 4.3.3 (Uniqueness of CP decomposition)**.** *An interesting property of higher-order tensors is that their rank decompositions are often unique, whereas matrix decompositions are not [139, 97, 35].*

Numerically, the first issue during the computation of the classical *CP* decomposition is the choice of R; the number of components. To remedy this problem, we fit multiple *CP* decompositions with different numbers of components until one is good. Let us consider the third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$. The fact of computing a CP decomposition with R component that best approximates $\mathcal{X}$ can be reformulated mathematically as follows

$$\min_{\widetilde{\mathcal{X}}} \|\mathcal{X} - \widetilde{\mathcal{X}}\| \quad \text{with} \quad \widetilde{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r a_r \circ b_r \circ c_r = [|\lambda, A, B, C|], \tag{4.12}$$

where $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_R]$ is defined by normalization.

The minimization problem (4.12) can be solve using an Alternating Least Squares (ALS) approach. By fixing B and C, solve the problem for A, then fixes A and C to solve the problem for B, then fixes A and B to solve for C, and continues to repeat the entire procedure until some convergence criterion is satisfied. By adopting such approach and having fixed all but one matrix, the problem reduces to a linear least-squares problem. For example, suppose that B and C are fixed. Then, the problem to solve became as follows

$$Find\ A\ such\ as\ \widetilde{\mathcal{X}}_{(1)} = A\ diag(\lambda)(C \odot B)^T.$$

In addition, by considering $\widehat{A} = A\ diag(\lambda)$, we can rewrite the above minimization problem in matrix form as

$$\min_{\widehat{A}} \|\mathcal{X}_{(1)} - \widehat{A}(C \odot B)^T\|_F.$$

The optimal solution is then given by

$$\widehat{A} = \mathcal{X}_{(1)}[(C \odot B)^T]^{\dagger}.$$

According to Proposition 4.2.1, the matrix $\widehat{A}$ is written as

$$\widehat{A} = \mathcal{X}_{(1)}(C \odot B)((C^T C) * (B^T B))^{\dagger}.$$

The advantage of using the above equation is that we need only calculate the pseudoinverse of an $R \times R$ matrix rather than a $JK \times R$ matrix. Finally, we normalize the columns of $\widehat{A}$ to get A. Namely, we compute λ by

$$\lambda_r = \|\widetilde{a}_r\|, \ \ and \ \ a_r = \frac{\widetilde{a}_r}{\lambda_r}, \ \forall r = 1, \ldots, R.$$

CHAPTER 5

Tensorial Tikhonov conditional gradient method for solving multidimensional ill-posed problems

5.1 Introduction

Let us consider the following tensor equation given by

$$\mathcal{X} \times_1 H_1^{(1)} \times_2 H_2^{(1)} + \mathcal{X} \times_1 H_1^{(2)} \times_2 H_2^{(2)} + \cdots + \mathcal{X} \times_1 H_1^{(r)} \times_2 H_2^{(r)} = \mathcal{B}, \tag{5.1}$$

where the coefficient matrices $H_1^{(i)} \in \mathbb{R}^{n_1 \times n_1}$ and $H_2^{(i)} \in \mathbb{R}^{n_2 \times n_2}$ $(\forall i = 1, \ldots, r)$ are known, $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ is the unknown tensor to be determined. Typically, the right-hand side tensor $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$, is contaminated by an error $\mathcal{E}$, i.e.

$$\mathcal{B} = \tilde{\mathcal{B}} + \mathcal{E},$$

where $\tilde{\mathcal{B}}$ denotes the unavailable error-free right-hand side.

Recently, a lot of articles developed the global form of well-known iterative projection methods in tensor forms to solve a class of tensor equations via the n-mode tensor product [7, 6]. The iterative method in the present chapter is well suited to solve such ill-conditioned tensor problems by enforcing double regularization process, first by using the Tikhonov regularization techniques, which is discussed in several reviews such as [13, 74, 19, 16, 88], and by minimizing our main problem under a convex constraint. For an optimal choice of an regularization parameter, we developed the Generalized Cross Validation (GCV) technique [60].

In the remainder of this work and for the sake of clarity, we consider only

the simplified case where $r = 1$. Then, the tensorial inverse problem is expressed as follows

$$\mathcal{X} \times_1 H_1 \times_2 H_2 = \tilde{\mathcal{B}} + \mathcal{E}. \tag{5.2}$$

5.2 Tensorial Tikhonov regularization problem

The constrained tensorial Tikhonov regularization problem associated with the tensor equation (5.2) can be written as follows

$$\min_{\mathcal{X}\in\Omega} \|\mathcal{X} \times_1 H_1 \times_2 H_2 - \mathcal{B}\|_F^2 + \mu^2 \|\mathcal{X} \times_1 R_1 \times_2 R_2\|_F^2, \tag{5.3}$$

where H_1 and H_2 represent the coefficient matrices from the tensorial inverse problem (5.2), R_1 and R_2 denote the regularization matrices, the set Ω is a convex constraint over $\mathcal{X}$, and μ is the regularization parameter. The convex set Ω can be, for example, a closed ball or a box. By searching our solution in such subset, we control the recovery of the edges.

Let us consider the following linear operation

$$\begin{aligned} \mathscr{H} : \mathbb{R}^{n_1\times\cdots\times n_N} &\longrightarrow \mathbb{R}^{n_1\times\cdots\times n_N} \\ \mathcal{X} &\longrightarrow \mathcal{X} \times_1 H_1 \times_2 H_2, \end{aligned}$$

$$\begin{aligned} \mathscr{R} : \mathbb{R}^{n_1\times\cdots\times n_N} &\longrightarrow \mathbb{R}^{n_1\times\cdots\times n_N} \\ \mathcal{X} &\longrightarrow \mathcal{X} \times_1 R_1 \times_2 R_2. \end{aligned}$$

By simple algebraic manipulations using *n-mode product* properties 4.2.2, we can obtain the adjoint operators $\mathscr{H}^*$ and $\mathscr{R}^*$ of $\mathscr{H}$ and $\mathscr{R}$, respectively,

$$\begin{aligned} \mathscr{H}^* : \mathbb{R}^{n_1\times\cdots\times n_N} &\longrightarrow \mathbb{R}^{n_1\times\cdots\times n_N} \\ \mathcal{X} &\longrightarrow \mathcal{X} \times_1 H_1^T \times_2 H_2^T, \end{aligned}$$

$$\begin{aligned} \mathscr{R}^* : \mathbb{R}^{n_1\times\cdots\times n_N} &\longrightarrow \mathbb{R}^{n_1\times\cdots\times n_N} \\ \mathcal{X} &\longrightarrow \mathcal{X} \times_1 R_1^T \times_2 R_2^T. \end{aligned}$$

The convex set $\Omega \subset \mathbb{R}^{n_1\times\cdots\times n_N}$ can be a closed ball of radius r and center 0,

$$\Omega = \bar{B}_r = \{\mathcal{X} \in \mathbb{R}^{n_1\times\cdots\times n_N} \mid \|\mathcal{X}\|_F \leqslant r\}, \tag{5.4}$$

or a box,

$$\Omega = B(\mathcal{L},\mathcal{U}) = \{\mathcal{X} \in \mathbb{R}^{n_1\times\cdots\times n_N} \mid \mathcal{L} \leqslant \mathcal{X} \leqslant \mathcal{U}\}, \tag{5.5}$$

where $\mathcal{L}$ and $\mathcal{U}$ are two given tensors and the inequality $\mathcal{L} \leqslant \mathcal{X}$ between two

Nth-order tensors $\mathcal{L}$ and $\mathcal{X}$ means $\mathcal{L}_{i_1,\ldots,i_N} \leqslant \mathcal{X}_{i_1,\ldots,i_N}$ for all $i_1,\ldots,i_N$.

Now, by considering the following objective function,

$$\begin{array}{rcl} \mathcal{J}_\mu : \mathbb{R}^{n_1\times\cdots\times n_N} & \longrightarrow & \mathbb{R}_+ \\ \mathcal{X} & \longrightarrow & \mathcal{J}_\mu(\mathcal{X}) = \|\mathscr{H}(\mathcal{X}) - \mathcal{B}\|_F^2 + \mu^2\|\mathscr{R}(\mathcal{X})\|_F^2. \end{array} \tag{5.6}$$

the tensorial convex constrained problem (5.3) is written as

$$\min_{\mathcal{X}\in\Omega} \mathcal{J}_\mu(\mathcal{X}). \tag{5.7}$$

It is clear that the objective function is differentiable and its gradient is computed in the following proposition. In addition, if the set Ω is further compact, then this minimization problem has a unique solution.

Proposition 5.2.1. *The gradient of $\mathcal{J}_\mu$ at $\mathcal{X} \in \mathbb{R}^{n_1\times\cdots\times n_N}$ is given by*

$$\nabla\mathcal{J}_\mu(\mathcal{X}) = 2\Big[\mathscr{H}^*(\mathscr{H}(\mathcal{X}) - \mathcal{B}) + \mu^2(\mathscr{R}^*(\mathscr{R}(\mathcal{X}))\Big]. \tag{5.8}$$

Proof. Consider the function g_μ defined by,

$$\begin{array}{rcl} g_\mu : \mathbb{R} & \longrightarrow & \mathbb{R} \\ t & \longrightarrow & g_\mu(t) = \mathcal{J}_\mu(\mathcal{X} + t\mathcal{K}), \text{ for all } \mathcal{K}. \end{array} \tag{5.9}$$

It is clear that $g_\mu^{'}(0) = \langle\nabla\mathcal{J}_\mu(\mathcal{X}), \mathcal{K}\rangle_F$. Furthermore, we have

$$\begin{aligned} g_\mu^{'}(0) &= 2\langle\mathscr{H}(\mathcal{X}) - \mathcal{B}, \mathscr{H}(\mathcal{K})\rangle_F + 2\mu^2\langle\mathscr{R}(\mathcal{X}), \mathscr{R}(\mathcal{K})\rangle_F \\ &= 2\langle\mathscr{H}^*(\mathscr{H}(\mathcal{X}) - \mathcal{B}) + \mu^2\mathscr{R}^*(\mathscr{R}(\mathcal{X})), \mathcal{K}\rangle_F. \end{aligned}$$

As a consequence, the gradient of $\mathcal{J}_\mu$ at $\mathcal{X}$ is then given by

$$\nabla\mathcal{J}_\mu(\mathcal{X}) = 2\Big[\mathscr{H}^*(\mathscr{H}(\mathcal{X}) - \mathcal{B}) + \mu^2\mathscr{R}^*(\mathscr{R}(\mathcal{X}))\Big]. \tag{5.10}$$

In addition, we can prove that

$$\mathcal{J}_\mu(\mathcal{X} + \mathcal{K}) = \mathcal{J}_\mu(\mathcal{X}) + \langle\nabla\mathcal{J}_\mu(\mathcal{X}), \mathcal{K}\rangle_F + o(\|\mathcal{K}\|_F). \tag{5.11}$$

Indeed, by applying Taylor's formula on g_μ, we got

$$g_\mu(1) = g_\mu(0) + g_\mu^{'}(0) + \frac{1}{2}g_\mu^{''}(0),$$

which implies

$$\begin{aligned} |o(\|\mathcal{K}\|_F)| &= \left|\frac{1}{2} g''_\mu(0)\right| \\ &\leqslant \|\mathscr{H}(\mathcal{K})\|_F^2 + \mu^2 \|\mathscr{R}(\mathcal{K})\|_F^2 \\ &\leqslant M\|\mathcal{K}\|_F^2, \end{aligned} \tag{5.12}$$

where $M = \|H_1\|_F^2\|H_2\|_F^2 + \mu^2\|R_1\|_F^2\|R_2\|_F^2$, which verifies the equation (5.11).

□

The treatment of the minimization problem (5.3) differs according to the presence or absence of the constraint Ω. In the unconstrained case, it is immediate to prove that the problem is equivalent to a generalized tensorial Sylvester equation that we may solve using some Krylov subspace methods.

The Krylov subspace methods are considered the most efficient methods used to solve a large linear equation. These methods are based on the structure and the proprieties of the Krylov subspace developed by Russian mathematician and naval engineer, Alexei Krylov, in 1931 [99]. The idea behind the construction of its subspaces comes from the Cayley-Hamilton theorem, which says that the inverse of a matrix A is expressed in terms of a linear combination of powers of A[136]. Thereafter, Krylov subspaces are used in algorithms for finding approximate solutions to linear algebra problems [136]. These iterative approaches search for a solution in the Krylov subspace associated with the linear equation starting from an initial approximation and the corresponding residual.

Among the best known methods, Generalized minimal residual method (GMRES) is an iterative Krylov subspace method for solving indefinite nonsymmetric system of linear equations. The method approximates the solution by the approximation in a Krylov subspace with minimal residual. The Arnoldi iteration is used to find this approximation by computing the Krylov subspace basis. The *GMRES* method was developed by Yousef Saad and Martin H. Schultz in 1986 [135] as a generalization and improvement of Minimal residual method (MINRES) [125] developed for symmetric matrices. We found also the Lanczos algorithm for Hermitian matrices [136], the Least-squares method (LSQR) based on Golub Khan bidiagonalization [124] and others. The Block versions of all these techniques are developed also to solve systems of linear equations with multiple right-hand sides [86, 50]. Recently, the Krylov subspace methods are generalized in the tensor form [6, 64, 65].

5.3 Tensorial Tikhonov conditional gradient method

In the presence of a convex constraint Ω, the conditional gradient method will be used to solve the following tensorial constrained Tikhonov regularization problem,

$$\min_{\mathcal{X}\in\Omega}\Big(\|\mathscr{H}(\mathcal{X})-\mathcal{B}\|_F^2+\mu^2\|\mathscr{R}(\mathcal{X})\|_F^2\Big). \tag{5.13}$$

This method was first introduced in [49] as one of successful gradient descent algorithms used to solve nonlinear optimization problems, and used after in several papers [19, 18, 11]. The idea of the method is to approximate the solution of our minimization problem by a sequence $(\mathcal{X}_k)_{k\in\mathbb{N}}$ defined as

$$\mathcal{X}_{k+1}=\mathcal{X}_k+\alpha_k\mathcal{D}_k,$$

for a given initial guess $\mathcal{X}_0$. It is clear that the effectiveness of the approach is based on an appropriate choice of the direction $\mathcal{D}_k$ and the step size α_k.

Suppose that we have a given iterate tensor $\mathcal{X}_k\in\Omega$ at the step k, we consider the first-order Taylor polynomial to the objective function $\mathscr{J}_\mu$ (see (5.6)) at $\mathcal{X}_k$

$$\mathscr{J}_\mu(\mathcal{X})\approx\mathscr{J}_\mu(\mathcal{X}_k)+\langle\nabla\mathscr{J}_\mu(\mathcal{X}_k),\mathcal{X}-\mathcal{X}_k\rangle+o(\|\mathcal{X}-\mathcal{X}_k\|_F).$$

Then, our minimization problem (5.13) is approached, at each step k, by the following linear optimization problem

$$\bar{\mathcal{X}_k}^*=\operatorname*{argmin}_{\mathcal{X}\in\Omega}\langle\nabla\mathscr{J}_\mu(\mathcal{X}_k),\mathcal{X}\rangle. \tag{5.14}$$

Since Ω is a convex set, the straight line segment $[\mathcal{X}_k,\bar{\mathcal{X}_k}^*]$ lies entirely inside Ω and the tensor $\mathcal{D}_k=\bar{\mathcal{X}_k}^*-\mathcal{X}_k$ is a feasible direction.

On the other hand, the step size α_k is defined as the minimizer of

$$\min_{0\leqslant\alpha\leqslant1}\mathscr{J}_\mu(\mathcal{X}_k+\alpha\mathcal{D}_k). \tag{5.15}$$

As a result, the tensorial conditional gradient algorithm can be summarized as in Algorithm 5.1.

5.3.1 Direction $\mathcal{D}_k$

The choice of the direction $\mathcal{D}_k$ in each step k, depends on the solution of (5.14) which is depending on the choice of the convex constraint subset Ω. The following proposition gives the iterate $\bar{\mathcal{X}_k}^*$ for two examples of Ω.

Algorithm 5.1 Tensorial Conditional Gradient Algorithm

1: Choose a tolerance ϵ, an initial guess $\mathcal{X}_0 \in \Omega$, an integer k_{max}, and set $k = 0$.
2: **while** $\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F / \|\mathcal{X}_k\|_F \geqslant \epsilon$ **and** $k \leqslant k_{max}$ **do**
3: Solve the minimization problem over the set Ω:

$$\bar{\mathcal{X}_k}^* = \underset{\mathcal{X}\in\Omega}{\operatorname{argmin}} \langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \mathcal{X} \rangle. \tag{5.16}$$

4: Compute the value: $\rho_k = \langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \bar{\mathcal{X}_k}^* - \mathcal{X}_k \rangle$.
5: **if** $|\rho_k| < \epsilon$ **then**
6: Set $\mathcal{D}_k = \bar{\mathcal{X}_k}^* - \mathcal{X}_k$ and solve the one dimensional minimization problem to find the step size:

$$\alpha_k^* = \underset{0\leqslant\alpha\leqslant 1}{\operatorname{argmin}} \mathcal{J}_\mu(\mathcal{X}_k + \alpha \mathcal{D}_k).$$

7: **end if**
8: Update $\mathcal{X}_{k+1} = \mathcal{X}_k + \alpha_k^* \mathcal{D}_k$, and set $k = k + 1$.
9: **end while**

Proposition 5.3.1. *At the iteration k, the solution $\bar{\mathcal{X}_k}^*$ of the minimization problem (5.16) with compact constraint is given by:*

i. If the set $\Omega = B(\mathcal{L}, \mathcal{U})$ is a box , then the iterate $\bar{\mathcal{X}_k}^*$ can be expressed as:

$$\forall (i_1, \ldots, i_N), \quad (\bar{\mathcal{X}_k}^*)_{i_1,\ldots,i_N} = \begin{cases} \mathcal{L}_{i_1,\ldots,i_N} & if \quad (\nabla \mathcal{J}_\mu(\mathcal{X}_k))_{i_1,\ldots,i_N} \geqslant 0 \\ \mathcal{U}_{i_1,\ldots,i_N} & if \quad (\nabla \mathcal{J}_\mu(\mathcal{X}_k))_{i_1,\ldots,i_N} < 0. \end{cases} \tag{5.17}$$

ii. If the set $\Omega = \bar{B}_r$ is a closed (metric) ball of center 0 and radius r, then $\bar{\mathcal{X}_k}^*$ is written as:

$$\bar{\mathcal{X}_k}^* = -r \frac{\nabla \mathcal{J}_\mu(\mathcal{X}_k)}{\|\mathcal{J}_\mu(\mathcal{X}_k)\|_F}. \tag{5.18}$$

Proof. The value of $\bar{\mathcal{X}_k}^*$ is based on the set Ω,

i. If the convex set Ω consists of a box $B(\mathcal{L}, \mathcal{U})$, then, $\bar{\mathcal{X}_k}^*$ given by (5.17) leads to the following inequality: for all $\mathcal{X} \in B(\mathcal{L}, \mathcal{U})$,

$$(\nabla \mathcal{J}_\mu(\mathcal{X}_k))_{i_1,\ldots,i_N} (\bar{\mathcal{X}_k}^*)_{i_1,\ldots,i_N} \leqslant (\nabla \mathcal{J}_\mu(\mathcal{X}_k))_{i_1,\ldots,i_N} (\mathcal{X})_{i_1,\ldots,i_N}.$$

and

$$\langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \bar{\mathcal{X}_k}^* \rangle \leqslant \langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \mathcal{X} \rangle, \quad \forall \mathcal{X} \in B(\mathcal{L}, \mathcal{U}).$$

ii. If the convex set Ω consists of the closed ball $\bar{B}_r$, then, for all $\mathcal{X} \in \bar{B}_r$, the Cauchy-Schwarz inequality leads to

$$\begin{aligned}\langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \mathcal{X} \rangle &\geqslant -\|\nabla \mathcal{J}_\mu(\mathcal{X}_k)\|_F \|\mathcal{X}\|_F \\ &\geqslant -r\|\nabla \mathcal{J}_\mu(\mathcal{X}_k)\|_F = \left\langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), -r\frac{\nabla \mathcal{J}_\mu(\mathcal{X}_k)}{\|\nabla \mathcal{J}_\mu(\mathcal{X}_k)\|_F} \right\rangle_F,\end{aligned}$$

then, it immediate to see that,

$$\langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \bar{\mathcal{X}_k}^* \rangle \leqslant \langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \mathcal{X} \rangle, \quad \forall \mathcal{X} \in \bar{B}_r,$$

where $\bar{\mathcal{X}_k}^*$ is given by (5.18).

□

5.3.2 Step size α_k

At the iteration k, the step size α_k is defined as the minimizer of the functional $\phi(\alpha) = \mathcal{J}_\mu(\mathcal{X}_k + \alpha \mathcal{D}_k)$ which mean:

$$\alpha_k = \underset{0 \leqslant \alpha \leqslant 1}{\operatorname{argmin}} \mathcal{J}_\mu(\mathcal{X}_k + \alpha \mathcal{D}_k). \tag{5.19}$$

The previous minimization problem leads to a polynomial equation using the definition of $\mathcal{J}_\mu$, which make the problem (5.19) equivalent to:

$$\alpha_k = \underset{0 \leqslant \alpha \leqslant 1}{\operatorname{argmin}} \left(a_k \alpha^2 + b_k \alpha + c_k \right), \tag{5.20}$$

where

$$\begin{aligned} a_k &= \|\mathscr{H}(\mathcal{D}_k)\|_F^2 + \mu^2 \|\mathscr{R}(\mathcal{D}_k)\|_F^2, \\ b_k &= \langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \mathcal{D}_k \rangle, \\ c_k &= \mathcal{J}_\mu(\mathcal{X}_k). \end{aligned}$$

At the iteration k, the solution of the problem (5.20) is expressed as

$$\alpha_k = P_{[0,1]}(\tilde{\alpha_k}), \tag{5.21}$$

where $P_{[0,1]} : \mathbb{R} \longrightarrow [0,1]$ is the orthogonal projection on the segment $[0,1]$, and

$\tilde{\alpha}_k := \underset{\tilde{\alpha}}{\operatorname{argmin}}\left(a_k\tilde{\alpha}^2 + b_k\tilde{\alpha} + c_k\right)$. Since $a_k > 0$, then, the scaler $\tilde{\alpha}_k$ is given by

$$\tilde{\alpha}_k = -\frac{1}{2}\frac{\langle \nabla \mathscr{J}_\mu(\mathcal{X}_k), \mathcal{D}_k\rangle}{\|\mathscr{H}(\mathcal{D}_k)\|_F^2 + \mu^2\|\mathscr{R}(\mathcal{D}_k)\|_F^2}.$$

According to the problem (5.14), we have $\langle \nabla \mathscr{J}_\mu(\mathcal{X}_k), \mathcal{D}_k\rangle \leqslant 0$ showing that $\tilde{\alpha}_k \geqslant 0$, then, the step size α_k is given by

$$\alpha_k = \begin{cases} \tilde{\alpha}_k & if \quad 0 \leqslant \tilde{\alpha}_k \leqslant 1 \\ 1 & if \quad \tilde{\alpha}_k > 1 \end{cases}. \tag{5.22}$$

5.3.3 Convergence results

Theorem 5.3.1. *Let ρ_k be a sequence given by $\rho_k = \langle \nabla \mathscr{J}_\mu(\mathcal{X}_k), \bar{\mathcal{X}}_k^* - \mathcal{X}_k\rangle$, where $\bar{\mathcal{X}}_k^*$ is a solution of the minimization problem (5.14), then*

$$\lim_{k\to+\infty} \rho_k = 0. \tag{5.23}$$

Proof. For any $\alpha \in [0,1]$, we set

$$\mathcal{X}_k(\alpha) = \mathcal{X}_k + \alpha(\bar{\mathcal{X}}_k^* - \mathcal{X}_k).$$

According to Taylor formula in (5.11), it follows that

$$\mathscr{J}_\mu(\mathcal{X}_k(\alpha)) - \mathscr{J}_\mu(\mathcal{X}_k) = \alpha\langle \nabla \mathscr{J}_\mu(\mathcal{X}_k), \bar{\mathcal{X}}_k^* - \mathcal{X}_k\rangle + o(\alpha\|\bar{\mathcal{X}}_k^* - \mathcal{X}_k\|).$$

Furthermore, the equation (5.12) shows that

$$o(\alpha\|\bar{\mathcal{X}}_k^* - \mathcal{X}_k\|) \leqslant \alpha^2 M\|\bar{\mathcal{X}}_k^* - \mathcal{X}_k\|^2 \leqslant \alpha^2 MC^2,$$

where the constant $C := \max_{\mathcal{X},\mathcal{Y}\in\Omega} \|\mathcal{X} - \mathcal{Y}\|$ is the diameter of the compact set Ω. As a consequence, we get

$$\mathscr{J}_\mu(\mathcal{X}_k(\alpha)) - \mathscr{J}_\mu(\mathcal{X}_k) \leqslant \alpha\rho_k + \alpha^2 MC^2.$$

On the other hand, using the fact that $\mathcal{X}_{k+1} = \mathcal{X}_k(\alpha_k)$, with α_k is the step size given by (5.22), it follows that

$$\mathscr{J}_\mu(\mathcal{X}_{k+1}) \leqslant \mathscr{J}_\mu(\mathcal{X}_k(\alpha)), \quad \forall \alpha \in [0,1].$$

Then, we get

$$\mathcal{J}_\mu(\mathcal{X}_{k+1}) - \mathcal{J}_\mu(\mathcal{X}_k) \leqslant \alpha\rho_k + \alpha^2 MC^2. \tag{5.24}$$

Since $\bar{\mathcal{X}_k}^*$ is the solution of (5.14), we have $\rho_k = \langle \nabla\mathcal{J}_\mu(\mathcal{X}_k), \bar{\mathcal{X}_k}^* - \mathcal{X}_k\rangle \leqslant 0$, then,

$$\mathcal{J}_\mu(\mathcal{X}_k) - \mathcal{J}_\mu(\mathcal{X}_{k+1}) \geqslant -\alpha\rho_k - \alpha^2 MC^2 = \alpha|\rho_k| - \alpha^2 MC^2. \tag{5.25}$$

In addition, for $\alpha = 0$, the inequality (5.24) leads to

$$\mathcal{J}_\mu(\mathcal{X}_{k+1}) \leqslant \mathcal{J}_\mu(\mathcal{X}_k),$$

which mean that the sequence $(\mathcal{J}_\mu(\mathcal{X}_k))_k$ is monotonically decreasing, in the other hand, the sequence is also bounded below on Ω , due to

$$\forall k, \quad \mathcal{J}_\mu(\mathcal{X}_k) \geqslant \min_{\mathcal{X}\in\Omega} \mathcal{J}_\mu(\mathcal{X}),$$

then, $(\mathcal{J}_\mu(\mathcal{X}_k))_k$ is convergent. Consequently,

$$\lim_{k\to+\infty} \mathcal{J}_\mu(\mathcal{X}_k) - \mathcal{J}_\mu(\mathcal{X}_{k+1}) = 0.$$

Using (5.25), it follows that

$$0 < |\rho_k| \leqslant \frac{\mathcal{J}_\mu(\mathcal{X}_k) - \mathcal{J}_\mu(\mathcal{X}_{k+1})}{\alpha} + \alpha MC^2. \tag{5.26}$$

If we take a limit in (5.26) for $k \to +\infty$, we obtain

$$0 \leqslant \liminf_{k\to+\infty} |\rho_k| \leqslant \limsup_{k\to+\infty} |\rho_k| \leqslant \alpha MC^2,$$

for all $\alpha \in]0,1[$. Now, if we take again a limit in the last inequality for $\alpha \to 0$, we get the desired result. □

Theorem 5.3.2. *The sequence generated by Algorithm 5.1, is a minimizing sequence of the functional $\mathcal{J}_\mu$ on the set Ω which mean*

$$\lim_{k\to+\infty} \mathcal{J}_\mu(\mathcal{X}_k) = \mathcal{J}_\mu(\mathcal{X}^*). \tag{5.27}$$

Proof. For any $\alpha \in [0,1]$, we set

$$\mathcal{X}_k(\alpha) = \mathcal{X}_k + \alpha(\bar{\mathcal{X}_k}^* - \mathcal{X}_k).$$

From Algorithm 5.1, the iterative approximations $(\mathcal{X}_k)_k$ for the exact solution is

given by

$$\mathcal{X}_{k+1} = \mathcal{X}_k + \alpha_k(\bar{\mathcal{X}}_k^* - \mathcal{X}_k) = \mathcal{X}_k(\alpha_k),$$

where α_k is a solution of the problem (5.15). By the property of the convex function, we have

$$\mathcal{J}_\mu(\mathcal{X}^*) - \mathcal{J}_\mu(\mathcal{X}_k) \geqslant \langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \mathcal{X}^* - \mathcal{X}_k \rangle_F,$$

where $\mathcal{X}^* \in \Omega$ is a solution of the problem (5.3). It follows that

$$\begin{aligned} 0 \leqslant \mathcal{J}_\mu(\mathcal{X}_k) - \mathcal{J}_\mu(\mathcal{X}^*) &\leqslant -\langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \mathcal{X}^* - \mathcal{X}_k \rangle_F \\ &\leqslant -\min_{\mathcal{X}} \langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \mathcal{X} - \mathcal{X}_k \rangle_F \\ &\leqslant -\langle \nabla \mathcal{J}_\mu(\mathcal{X}_k), \bar{\mathcal{X}}_k^* - \mathcal{X}_k \rangle_F \\ &\leqslant |\rho_k|. \end{aligned}$$

Then, we have

$$0 \leqslant \mathcal{J}_\mu(\mathcal{X}_k) - \mathcal{J}_\mu(\mathcal{X}^*) \leqslant |\rho_k|.$$

According to Theorem 5.3.1, we obtain the desired result. □

5.4 Parameter selection method for tensorial Tikhonov regularization

For an optimal selection of the regularization parameter μ, we will generalize the *GCV* method as being a method which does not depend on a prior knowledge about the noise variance. According to the mode-3 matricization and based on the classic definition of Generalized Cross Validation method [60], the following definition gives the associated Tensorial GCV (t-GCV) function to the tensorial model.

Definition 5.4.1. *The t-GCV function associated to the tensorial minimization problem (5.3) is defined as*

$$\begin{aligned} \mathcal{G} : \mathbb{R}_+^* &\longrightarrow \mathbb{R}_+ \\ \mu &\longrightarrow \mathcal{G}(\mu) = \frac{\|\mathscr{H}(\mathcal{X}_\mu) - \mathcal{B}\|_F^2}{trace(I - \bar{H} M_\mu^{-1} \bar{H}^T)^2}, \end{aligned} \tag{5.28}$$

where $\mathcal{X}_\mu$ is the regularized solution of (5.3), the matrices $\bar{H} = I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes H_2 \otimes H_1$ and $M_\mu = (I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes H_2^T H_2 \otimes H_1^T H_1) + \mu^2 (I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes R_2^T R_2 \otimes R_1^T R_1)$.

Then the optimal parameter μ^* for the proposed tensorial regularization method is the one that minimizes the t-GCV function $\mathcal{G}$,

$$\mu^* = \underset{\mu>0}{\operatorname{argmin}}\, \mathcal{G}(\mu). \tag{5.29}$$

Based on the spectral decomposition of the coefficient matrices and the regularization matrices, the following proposition presents a simplified expression of the *t-GCV* function $\mathcal{G}$.

Proposition 5.4.1. *Suppose the Generalized Singular Value Decomposition (GSVD) [126, 61] of pairs (H_1, R_1) and (H_2, R_2) are given respectively by*

$$\begin{cases} U_1^T H_1 V = S_1 = diag(s_{1,1}, s_{2,1}, \ldots, s_{n_1,1}), & s_{i,1} \geqslant 0, \\ W_1^T R_1 V = C_1 = diag(c_{1,1}, c_{2,1}, \ldots, c_{n_1,1}), & c_{i,1} \geqslant 0, \end{cases} \tag{5.30}$$

and

$$\begin{cases} U_2^T H_2 Z = S_2 = diag(s_{1,2}, s_{2,2}, \ldots, s_{n_2,2}), & s_{i,2} \geqslant 0, \\ W_2^T R_2 Z = C_2 = diag(c_{1,2}, c_{2,2}, \ldots, c_{n_2,2}), & c_{i,2} \geqslant 0 \end{cases} \tag{5.31}$$

where U_1, U_2, W_1, and W_2 are orthogonal matrices, V and Z are nonsingular matrices. Let set $S = I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes S_2 \otimes S_1 = diag(s_1, s_2, \ldots, s_K)$, and $C = I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes C_2 \otimes C_1 = diag(c_1, c_2, \ldots, c_K)$ with $K = n_1 \cdot n_2 \cdot n_4 \cdots n_N$. Then, the expression of the t-GCV function associated with the regularization problem (5.3), is given by

$$\mathcal{G}(\mu) = \frac{\displaystyle\sum_{i=1}^{n_3} \sum_{j=1}^{K} \left(\frac{c_j^2}{s_j^2 + \mu^2 c_j^2} u_j^T b_i \right)^2}{\displaystyle\left(\sum_{j=1}^{K} \frac{c_j^2}{s_j^2 + \mu^2 c_j^2} \right)^2}, \tag{5.32}$$

where u_j for $j = 1, \ldots, K$ and b_i for $i = 1, \ldots, n_3$ are defined below in the proof.

Proof. The regularized solution $\mathcal{X}_\mu$ verifies the following normal equation associated to the unconstrained minimization problem in (5.3),

$$\mathcal{X}_\mu \times_1 H_1^T H_1 \times_2 H_2^T H_2 + \mu^2 \mathcal{X}_\mu \times_1 R_1^T R_1 \times_2 R_2^T R_2 = \mathcal{B} \times_1 H_1^T \times_2 H_2^T. \tag{5.33}$$

By applying the mode-3 matricization to the equation (5.33), we obtain

$$\mathcal{X}_{\mu(3)} M_\mu = \mathcal{B}_{(3)} \bar{H}, \tag{5.34}$$

where $\mathcal{B}_{(3)} \in \mathbb{R}^{n_3 \times K}$ represents the mode-3 matricization of the tensor $\mathcal{B}$ defined

by $\mathcal{B}_{(3)} = [b_1, \ldots, b_{n_3}]^T$, $\bar{H}$ and M_μ are $K \times K$ matrices defined respectively by

$$\bar{H} = I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes H_2 \otimes H_1,$$

$$M_\mu = (I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes H_2^T H_2 \otimes H_1^T H_1) + \mu^2 (I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes R_2^T R_2 \otimes R_1^T R_1).$$

Then, it is immediate to observe that

$$\begin{aligned} \|\mathscr{H}(\mathcal{X}_\mu) - \mathcal{B}\|_F^2 &= \|\mathcal{X}_\mu \times_1 H_1 \times_2 H_2 - \mathcal{B}\|_F^2 \\ &= \|\mathcal{X}_{\mu(3)} (I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes H_2 \otimes H_1)^T - \mathcal{B}_{(3)}\|_F^2 \\ &= \|\mathcal{B}_{(3)} \left(\bar{H} M_\mu^{-1} \bar{H}^T - I_K \right)\|_F^2 \end{aligned}$$

By using the *GSVD* decomposition of the pairs matrices (H_1, R_1) and (H_2, R_2) given by equations (5.30) and (5.31), the matrices $\bar{H}$ and M_μ can be simplified as follows

$$\bar{H} = U S Y^{-1}, \tag{5.35}$$

and

$$M_\mu = Y^{-T} (S^2 + \mu^2 C^2) Y^{-1}, \tag{5.36}$$

where $U = I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes U_2 \otimes U_1 = [u_1, u_2, \ldots, u_K]$ is $K \times K$ orthogonal matrix, $S = I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes S_2 \otimes S_1$ and $C = I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes C_2 \otimes C_1$ are diagonal matrices, and $Y = I_{n_N} \otimes \cdots \otimes I_{n_4} \otimes Z \otimes V$ is nonsingular matrix.

As a consequence, we get

$$\begin{aligned} \|\mathscr{H}(\mathcal{X}_\mu) - \mathcal{B}\|_F^2 &= \|\mathcal{B}_{(3)} U \left(S(S^2 + \mu^2 C^2)^{-1} S - I_K \right)\|_F^2 \\ &= \|\left(S(S^2 + \mu^2 C^2)^{-1} S - I_K \right) U^T \mathcal{B}_{(3)}^T\|_F^2 \\ &= \sum_{i=1}^{n_3} \|\left(S(S^2 + \mu^2 C^2)^{-1} S - I_K \right) U^T b_i\|_2^2 \\ &= \sum_{i=1}^{n_3} \sum_{j=1}^{K} \left[\left(S(S^2 + \mu^2 C^2)^{-1} S - I_K \right) U^T b_i \right]_j^2 \\ &= \sum_{i=1}^{n_3} \sum_{j=1}^{K} \left[\left(\frac{s_j^2}{s_j^2 + \mu^2 c_j^2} - 1 \right) u_j^T b_i \right]^2 \\ &= \sum_{i=1}^{n_3} \sum_{j=1}^{K} \left(\frac{\mu^2 c_j^2}{s_j^2 + \mu^2 c_j^2} u_j^T b_i \right)^2. \end{aligned}$$

On the other hand, by using the decompositions (5.35) and (5.36) , it is

immediate to prove that

$$trace(I_K - \bar{H}M_\mu^{-1}\bar{H}^T) = trace(I_K - S(S^2 + \mu^2 C^2)^{-1}S)$$
$$= \sum_{j=1}^{K} \frac{\mu^2 c_j^2}{s_j^2 + \mu^2 c_j^2}.$$

Finally, we get the expression of the *t-GCV* function as given in equation(5.32). □

We obtain the following algorithm describing the tensorial conditional gradient Tikhonov method *t-CGT*.

Algorithm 5.2 Tensorial Conditional Gradient Tikhonov(*t-CGT*).

1: Given $\mathcal{B}$, $\mathcal{X}_0$, $\mathscr{H}$, $\mathscr{R}$, Ω, ϵ, an integer k_{max}, and set $k = 0$.
2: Compute the parameter μ by minimizing the *t-GCV* function (5.32).
3: **while** ($\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F / \|\mathcal{X}_k\|_F \geqslant \epsilon$ **and** $k \leqslant k_{max}$) **do**
4: Compute $\bar{\mathcal{X}}_k^*$ depends on the choice of Ω using the proposition 5.3.1.
5: Set the direction of descent: $\mathcal{D}_k = \bar{\mathcal{X}}_k^* - \mathcal{X}_k$.
6: Compute the step size α_k verifies (5.22).
7: Compute $\mathcal{X}_{k+1} = \mathcal{X}_k + \alpha_k \mathcal{D}_k$,
8: **end while**

5.5 Conclusion

We presented in this chapter a new approach for solving multidimensional ill-posed problems regularized by Tikhonov penalty. The tensorial Tikhonov minimization problem is solved either using the tensorial global *GMRES* in the unconstrained case, or by using the conditional gradient method in the presence of a constraint Ω. In addition, we generalized the *GCV* method to compute the optimal regularization parameter. The Tikhonov regularization produces a smoothing effect on the resulting solution as it determines smooth features while blurring jumps in the regularized solution, a characteristic known as over-smoothing that can be considered a significant drawback of this method, mainly if we aim to preserve edges. In this case, an edge-preserving regularization technique should be used.

CHAPTER 6

Tensorial total variation regularization problem with optimized projection methods

6.1 Introduction

We are interesting in this chapter to the tensorial representation of the total variation method to regularize ill posed tensor equation. Tensorial total variation (TTV) regularization problem can be expressed as a non-linear minimization problem of the form

$$\min_{\mathcal{X}} \left(|||\mathscr{H}(\mathcal{X}) - \mathcal{B}|||_k^k + \mu TTV_p(\mathcal{X}) \right),\ k, p = 1, 2, \tag{6.1}$$

where $\mathscr{H}$ stands for a tensorial linear operation, TTV_p is the tensorial total variation term and μ denotes the regularization parameter. Unlike the Tikhonov regularization problem, the cost function in this case is non-linear and non-differentiable, which makes solving the problem more complex. For this reason, an alternating technique will be used to split the main problem until some feasible subproblems are obtained.

6.2 Tensorial total variation regularization model

The discrete form of the tensorial total variation term TTV_p is given by

$$TTV_p(\mathcal{X}) = \sum_{i_1,i_2,\ldots,i_N} \left(\sum_{n=1}^{N} |(\nabla_{(n)}\mathcal{X})_{i_1,i_2,\ldots,i_N}|^p \right)^{1/p}, \; p = 1,2, \tag{6.2}$$

where $(\nabla_{(n)}\mathcal{X})_n$ are the partial derivatives of $\mathcal{X}$ given by

$$\forall n = 1,\ldots,N\,, \quad (\nabla_{(n)}\mathcal{X})_{i_1,\ldots,i_N} = \begin{cases} \mathcal{X}_{i_1,\ldots,i_n+1,\ldots,i_N} - \mathcal{X}_{i_1,\ldots,i_n,\ldots,i_N} & if \quad i_n < I_n \\ 0 & if \quad i_n = I_n \end{cases}$$

The gradient of $\mathcal{X}$ can be defined as the column block tensor $\nabla\mathcal{X}$ of size $N \cdot I_1 \times \cdots \times I_N$ consisting of the partial derivatives $(\nabla_{(n)}\mathcal{X})_n$, i.e.

$$\nabla\mathcal{X} = \begin{bmatrix} \nabla_{(1)}\mathcal{X} \\ \vdots \\ \nabla_{(N)}\mathcal{X} \end{bmatrix}.$$

By using the properties of the n-mode product (see Chapter 4), the partial derivatives are given by the following proposition.

Proposition 6.2.1. *Let $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ be an Nth order tensor. The partial derivatives $(\nabla_{(n)}\mathcal{X})_n$ of $\mathcal{X}$ can be expressed as follows*

$$\forall 1 \leqslant n \leqslant N, \qquad \nabla_{(n)}\mathcal{X} = \mathcal{X} \times_n C_n, \tag{6.3}$$

where $C_n = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & 0 \end{pmatrix} \in \mathbb{R}^{I_n \times I_n}$. *Its transpose is given by*

$$\nabla^T_{(n)}\mathcal{X} = \mathcal{X} \times_n C_n^T.$$

Proof. Let n be an integer with $1 \leqslant n \leqslant N$. According to the definition of the n-mode product, we have for all $1 \leqslant j \leqslant I_n$,

$$(\mathcal{X} \times_n C_n)_{i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1,\ldots,i_N} c_{j,i_n}.$$

Based on the structure of the square matrix $C_n = (c_{i,j})_{i,j}$, it follows that

$$(\mathcal{X} \times_n C_n)_{i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N} = \begin{cases} \mathcal{X}_{i_1,\ldots,j+1,\ldots,i_N} - \mathcal{X}_{i_1,\ldots,j,\ldots,i_N} & if \quad j < I_n \\ 0 & if \quad j = I_n \end{cases}$$
$$= (\nabla_{(n)}\mathcal{X})_{i_1,\ldots,j,\ldots,i_N},$$

which establishes the formula. □

We say that TTV_p term is **isotropic**, when we set $p = 1$, and **anisotropic** when $p = 2$ [62]. The main difference between the properties of these two choices is that isotropic total variation is rotationally invariant [62, 29]. On the other hand, the minimization problem (6.1) is referred to as TTV_p/L^1, when $k = 1$, and as TTV_p/L^2 when $k = 2$. Finally, the existence and the uniqueness of the solution of the minimization problem (6.1) have ensured by supposing some conditions of regularity on the operator $\mathscr{H}$ and the second member $\mathcal{B}$ (see [67, 148] for more details).

Unfortunately, to solve both TTV/L^1 and TTV/L^2 models directly is very difficult due to the non-differentiability and non-linearity of the tensorial total variation term. For that purpose, we are going to use Alternating direction method for multipliers (ADMM) [51, 22]. The idea behind *ADMM* approach is to transform the unconstrained minimization problem into an equivalent constrained optimization problem, that will be splitted into easier and smaller subproblems using augmented Lagrangian method (ALM)[128]. These subproblems can be solved using some optimization properties and tensor projection methods such as *GMRES*, conjugate gradient method, and *LSQR* in an optimized way.

6.3 Tensorial total variation TTV_2/L^1 problem

Let us consider the following tensorial TTV_2/L^1 regularization problem

$$\min_{\mathcal{X}} \left(|||\mathscr{H}(\mathcal{X}) - \mathcal{B}|||_1 + \mu TTV_2(\mathcal{X}) \right), \tag{6.4}$$

where $\mathscr{H}$ stands for a tensorial linear operator. The tensorial total variation minimization problem (6.4) can be written as the following constrained minimization problem

$$\begin{cases} \min\limits_{\mathcal{R},\mathcal{Y}^{(1)},\ldots,\mathcal{Y}^{(N)}} \left(|||\mathcal{R}|||_1 + \mu \sum\limits_{i_1,\ldots,i_N} \left(\sum\limits_{n=1}^{N} (\mathcal{Y}^{(n)}_{i_1,\ldots,i_N})^2 \right)^{1/2} \right) \\ subject\ to \quad \mathcal{R} = \mathscr{H}(\mathcal{X}) - \mathcal{B} \quad and \quad \nabla_{(n)}\mathcal{X} = \mathcal{Y}^{(n)},\ \forall n. \end{cases} \tag{6.5}$$

This constrained minimization problem can be also formulated as follows

$$\begin{cases} \min\limits_{\mathcal{R},\mathcal{Y}} & (\mathscr{F}(\mathcal{R})+\mathscr{G}(\mathcal{Y})) \\ \textit{subject to} & \mathcal{R}=\mathscr{H}(\mathcal{X})-\mathcal{B} \quad \textit{and} \quad \nabla\mathcal{X}=\mathcal{Y}. \end{cases} \tag{6.6}$$

where we consider the functionals $\mathscr{G}(\mathcal{Y}) = \mu \sum\limits_{i_1,i_2,\ldots,i_N} \left(\sum\limits_{n=1}^{N} (\mathcal{Y}^{(n)}_{i_1,i_2,\ldots,i_N})^2 \right)^{1/2}$, and $\mathscr{F}(\mathcal{R}) = |||\mathcal{R}|||_1$, and the column block tensors $\nabla\mathcal{X} = [\nabla_{(1)}\mathcal{X}, \nabla_{(2)}\mathcal{X}, \ldots, \nabla_{(N)}\mathcal{X}]^T$ and $\mathcal{Y} = [\mathcal{Y}^{(1)}, \mathcal{Y}^{(2)}, \ldots, \mathcal{Y}^{(N)}]^T$.
The augmented Lagrangian function associating to (6.6) is defined as follow

$$\begin{aligned} \mathscr{L}_{\beta,\rho}(\mathcal{X},\mathcal{R},\mathcal{Y},\mathcal{V},\mathcal{W}) = \mathscr{F}(\mathcal{R})+\mathscr{G}(\mathcal{Y})+\langle\mathcal{V}|\mathscr{H}(\mathcal{X})-\mathcal{B}-\mathcal{R}\rangle+\langle\mathcal{W}|\nabla\mathcal{X}-\mathcal{Y}\rangle \\ +\frac{\beta}{2}\|\mathscr{H}(\mathcal{X})-\mathcal{B}-\mathcal{R}\|_F^2+\frac{\rho}{2}\|\nabla\mathcal{X}-\mathcal{Y}\|_F^2, \end{aligned}$$

where $\mathcal{V} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ and $\mathcal{W} \in \mathbb{R}^{N\cdot I_1\times\cdots\times I_N}$ are the Lagrange multipliers of the linear constraints, and $\beta,\rho > 0$ are the penalty parameters for the violation of these linear constraints. The convex problem (6.6) can be solved using the classical alternating direction method of multipliers (*ADMM*) that generates an iterative sequence $(\mathcal{X}_k,\mathcal{R}_k,\mathcal{Y}_k,\mathcal{V}_k,\mathcal{W}_k)_{k\geqslant 0}$ given as follows. For given initial guess $\mathcal{X}_0,\mathcal{R}_0,\mathcal{Y}_0,\mathcal{V}_0$ and $\mathcal{W}_0$, we consider the following subproblems

$$\begin{cases} 1. & \mathcal{Y}_{k+1} = \operatorname{argmin}\limits_{\mathcal{Y}} \mathscr{L}_{\beta,\rho}(\mathcal{X}_k,\mathcal{R}_k,\mathcal{Y},\mathcal{V}_k,\mathcal{W}_k), \\ 2. & \mathcal{R}_{k+1} = \operatorname{argmin}\limits_{\mathcal{R}} \mathscr{L}_{\beta,\rho}(\mathcal{X}_k,\mathcal{R},\mathcal{Y}_{k+1},\mathcal{V}_k,\mathcal{W}_k), \\ 3. & \mathcal{X}_{k+1} = \operatorname{argmin}\limits_{\mathcal{X}} \mathscr{L}_{\beta,\rho}(\mathcal{X},\mathcal{R}_{k+1},\mathcal{Y}_{k+1},\mathcal{V}_k,\mathcal{W}_k), \\ 4. & \mathcal{V}_{k+1} = \mathcal{V}_k+\beta(\mathscr{H}(\mathcal{X}_{k+1})-\mathcal{B}-\mathcal{R}_{k+1}), \\ 5. & \mathcal{W}_{k+1} = \mathcal{W}_k+\rho(\nabla\mathcal{X}_{k+1}-\mathcal{Y}_{k+1}). \end{cases} \tag{6.7}$$

Then the resolution of the main minimization problem consists of the resolution of three subproblems 1-3 in 6.7 followed by an update of the Lagrange multipliers $\mathcal{V}_k$ and $\mathcal{W}_k$. To solve $\mathcal{Y}$-subproblem and $\mathcal{R}$-subproblem, the Shrinkage formula will be used to produce the following results.

Proposition 6.3.1. *Given $\mathcal{X}$ and $\mathcal{R}$, computing the iterate $\mathcal{Y}_{k+1}$ from the problem (6.7) is equivalent to compute*

$$\mathcal{Y}^{(n)} = \max\left\{\mathcal{K}^{(n)}-\frac{\mu}{\rho},0\right\} sign(\mathcal{K}^{(n)}),\ \forall n, \tag{6.8}$$

where $\mathcal{K}^{(n)} = \nabla_{(n)}\mathcal{X} + \frac{1}{\rho}\mathcal{W}_k^{(n)}$.

Proof. The iterate $\mathcal{Y}_{k+1}$ can be obtained by solving the following minimization problem

$$\begin{aligned}
\mathcal{Y}_{k+1} &= \underset{\mathcal{Y}}{\operatorname{argmin}} \left(\mu \sum_{i_1,\ldots,i_N} \left(\sum_{n=1}^{N} (\mathcal{Y}_{i_1,\ldots,i_N}^{(n)})^2 \right)^{1/2} + \frac{\rho}{2} \|\nabla\mathcal{X} - \mathcal{Y}\|_F^2 + \langle \mathcal{W}_k | \nabla\mathcal{X} - \mathcal{Y} \rangle \right), \\
&= \underset{\mathcal{Y}^{(1)},\ldots,\mathcal{Y}^{(N)}}{\operatorname{argmin}} \left(\mu \sum_{i_1,\ldots,i_N} \left(\sum_{n=1}^{N} (\mathcal{Y}_{i_1,\ldots,i_N}^{(n)})^2 \right)^{1/2} + \frac{\rho}{2} \left\| \mathcal{Y} - \nabla\mathcal{X} - \frac{1}{\rho}\mathcal{W}_k \right\|_F^2 \right), \\
&= \underset{\mathcal{Y}^{(1)},\ldots,\mathcal{Y}^{(N)}}{\operatorname{argmin}} \left(\mu \sum_{i_1,\ldots,i_N} \left(\sum_{n=1}^{N} (\mathcal{Y}_{i_1,\ldots,i_N}^{(n)})^2 \right)^{1/2} + \frac{\rho}{2} \sum_{n=1}^{N} \left\| \mathcal{Y}^{(n)} - \nabla_{(n)}\mathcal{X} - \frac{1}{\rho}\mathcal{W}_k^{(n)} \right\|_F^2 \right), \\
&= \underset{\mathcal{Y}^{(1)},\ldots,\mathcal{Y}^{(N)}}{\operatorname{argmin}} \left[\sum_{i_1,\ldots,i_N} \left(\mu \left(\sum_{n=1}^{N} (\mathcal{Y}_{i_1,\ldots,i_N}^{(n)})^2 \right)^{1/2} + \frac{\rho}{2} \sum_{n=1}^{N} (\mathcal{Y}_{i_1,\ldots,i_N}^{(n)} - \mathcal{K}_{i_1,\ldots,i_N}^{(n)})^2 \right) \right],
\end{aligned}$$

where $\mathcal{K}^{(n)} = \nabla_{(n)}\mathcal{X} + \frac{1}{\rho}\mathcal{W}_k^{(n)}$. We remark that the subproblem is separable, then, it is equivalent to solve, for every $i_1,\ldots,i_N$, the minimization problem

$$\min_{\mathcal{Y}_{i_1,\ldots,i_N}} \left(\mu \left(\sum_{n=1}^{N} (\mathcal{Y}_{i_1,\ldots,i_N}^{(n)})^2 \right)^{1/2} + \frac{\rho}{2} \sum_{n=1}^{N} (\mathcal{Y}_{i_1,\ldots,i_N}^{(n)} - \mathcal{K}_{i_1,\ldots,i_N}^{(n)})^2 \right),$$

for which the unique minimizer is given by the following well-known two-dimensional shrinkage formula [106]:

$$\mathcal{Y}_{i_1,\ldots,i_N}^{(n)} = \max\left\{ \mathcal{K}_{i_1,\ldots,i_N}^{(n)} - \frac{\mu}{\rho}, 0 \right\} sign(\mathcal{K}_{i_1,\ldots,i_N}^{(n)}) \quad \forall n, \forall i_1,\ldots,i_N. \tag{6.9}$$

which is exactly

$$\mathcal{Y}^{(n)} = \max\left\{ \mathcal{K}^{(n)} - \frac{\mu}{\rho}, 0 \right\} sign(\mathcal{K}^{(n)}) \quad \forall n.$$

□

According to the *ADMM* process, the iterate $\mathcal{R}_{k+1}$ is defined as the solution of the following minimization problem

$$\min_{\mathcal{R}} \left(|||\mathcal{R}|||_1 + \langle \mathcal{V}_k | \mathscr{H}(\mathcal{X}) - \mathcal{B} - \mathcal{R} \rangle + \frac{\beta}{2} \|\mathscr{H}(\mathcal{X}) - \mathcal{B} - \mathcal{R}\|_F^2 \right). \tag{6.10}$$

Similarly to $\mathcal{Y}$ subproblem, it is immediate to prove the following result using one-dimensional Shrinkage formula [106].

Proposition 6.3.2. *Given $\mathcal{X}$ and $\mathcal{Y}$, the iterate $\mathcal{R}_{k+1}$ can be obtained by computing*

$$\mathcal{R} = \max\left\{|\mathscr{H}(\mathcal{X}) - \mathcal{B} + \frac{1}{\beta}\mathcal{V}_k| - \frac{1}{\beta}, 0\right\} sign(\mathscr{H}(\mathcal{X}) - \mathcal{B} + \frac{1}{\beta}\mathcal{V}_k), \tag{6.11}$$

where the absolute value operation $|\,.\,|$ and the sign operation $sign(.)$ are defined in an element-wise manner.

On the other hand, given $\mathcal{R}$ and $\mathcal{Y}$, the iterate $\mathcal{X}_{k+1}$ is obtained by solving the minimization problem

$$\min_{\mathcal{X}}\left(\langle \mathcal{V}_k|\mathscr{H}(\mathcal{X}) - \mathcal{B} - \mathcal{R}\rangle + \langle \mathcal{W}_k|\nabla\mathcal{X} - \mathcal{Y}\rangle + \frac{\beta}{2}\|\mathscr{H}(\mathcal{X}) - \mathcal{B} - \mathcal{R}\|_F^2 + \frac{\rho}{2}\|\nabla\mathcal{X} - \mathcal{Y}\|_F^2\right). \tag{6.12}$$

Due to the linearity and the differentiability of this minimization subproblem, we can update different projection methods to solve (6.12). We outline in the next paragraphs some interesting classical projection methods in a developed optimized form.

6.3.1 GMRES-BTF optimized algorithm

Using Euler-Lagrangian formula, the equation (6.12) is equivalent to the following normal equation

$$\rho\mathscr{H}^T(\mathscr{H}(\mathcal{X})) + \beta\nabla^T(\nabla(\mathcal{X})) = \beta\mathscr{H}^T(\mathcal{B}) + \mathscr{H}^T(\rho\mathcal{R} - \mathcal{V}_k) + \nabla^T(\beta\mathcal{Y} - \mathcal{W}_k). \tag{6.13}$$

Considering the symmetric operator $\mathscr{A}$ defined as

$$\begin{array}{rcl} \mathscr{A} : \mathbb{R}^{I_1\times\cdots\times I_N} & \longrightarrow & \mathbb{R}^{I_1\times\cdots\times I_N} \\ \mathcal{X} & \longrightarrow & \mathscr{A}(\mathcal{X}) = \rho\mathscr{H}^T(\mathscr{H}(\mathcal{X})) + \beta\nabla^T(\nabla(\mathcal{X})) \end{array} \tag{6.14}$$

and the second member $\mathcal{G}_k = \beta\mathscr{H}^T(\mathcal{B}) + \mathscr{H}^T(\rho\mathcal{R} - \mathcal{V}_k) + \nabla^T(\beta\mathcal{Y} - \mathcal{W}_k)$, the equation (6.13) is simplified to the following form

$$\mathscr{A}(\mathcal{X}) = \mathcal{G}_k,\ \forall k \geqslant 0. \tag{6.15}$$

Since the second member change at each step, the total tensor variation approach required solving the tensor equation at each step k. As we have seen previously, solving a linear equation by a Krylov subspace method consists of two main points: the computation of the Krylov subspace basis and the resolution of the projected matrix problem associated to the tensor equation. The tensor basis computation at each step k will definitely increase the cost of computation and affect the speed of the algorithm. For that purpose, instead of computing the krylov bases at each iteration,we will calculate

only the base in the first iteration and then we will complete the bases of the following iterations with a single element.

First, let us solve the linear equation $\mathcal{A}(\mathcal{X}_1) = \mathcal{G}_0$ using *GMRES-BTF*, which means constructing an F-orthonormal basis $\mathbb{V}_m = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_m\}$ of the following tensor Krylov subspace

$$\mathcal{K}_m(\mathscr{A}, \mathcal{P}_0) = span\{\mathcal{P}_0, \mathscr{A}(\mathcal{P}_0), \ldots, \mathscr{A}^{m-1}(\mathcal{P}_0)\},$$

where $\mathcal{P}_0 = \mathcal{G}_0 - \mathcal{A}(\mathcal{X}_0)$ is the residual, and $\mathcal{X}_0$ is an initial guess of $\mathcal{X}_1$. Then, it follows easily that

$$\mathscr{A}(\mathcal{V}_j) = \mathbb{V}_m \bar{\times}_{N+1} H_m(:, j), \ \forall j = 1, \ldots, m, \tag{6.16}$$

where $H_m \in \mathbb{R}^{(m+1)\times m}$ is an upper Hessenberg matrix and $\mathbb{V}_m$ is a $N+1$ order tensor of size $I_1 \times \cdots \times I_N \times m$ such that $\mathbb{V}_m(:, \ldots, :, j) = \mathcal{V}_j$ for all $j = 1, \ldots, m$. We search an approximated solution of $\mathcal{X}_1$ that satisfies

$$\mathcal{X}_1^{(m)} = \mathcal{X}_0 + \mathbb{V}_m \bar{\times}_{N+1} y^{(m)}, \tag{6.17}$$

where $y^{(m)}$ is the solution of the following reduced minimization problem

$$\min_y \|\beta e_1 - H_m y\|_2.$$

On the other hand, using the ***vec*** operator, the relations (6.16) and (6.17) will be in the following matrix forms, respectively,

$$[\boldsymbol{vec}(\mathscr{A}(\mathcal{V}_1)), \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_2)), \ldots, \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_m))] = V_m H_m, \tag{6.18}$$

and

$$x = x_0 + V_m y^{(m)}, \tag{6.19}$$

where $x = \boldsymbol{vec}(\mathcal{X})$, $x_0 = \boldsymbol{vec}(\mathcal{X}_0)$, and $V_m = [\boldsymbol{vec}(\mathcal{V}_1), \boldsymbol{vec}(\mathcal{V}_2), \ldots, \boldsymbol{vec}(\mathcal{V}_m)]$. In the same spirit of developing an optimized approach, we compute the QR factorization of $[\boldsymbol{vec}(\mathscr{A}(\mathcal{V}_1)), \ldots, \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_m))]$ in the first step, namely,

$$[\boldsymbol{vec}(\mathscr{A}(\mathcal{V}_1)), \ldots, \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_m))] = Q_m R_m, \tag{6.20}$$

where $Q_m \in \mathbb{R}^{I_1 \cdots I_N \times m}$ is an orthogonal matrix and $R_m \in \mathbb{R}^{m \times m}$ is an upper triangular matrix.

Once we apply the basic process in the first iteration, we will adopt a completion technique in the following iterations, to complete both the krylov subspace basis as well as the QR factorization. We now turn to the solution of

$$\mathscr{A}(\mathcal{X}) = \mathcal{G}_k, \quad k = 1, 2, \ldots$$

without repeating the same process in each iteration, we are going to use the F-orthonormal basis and the solution of the previous iteration. For example, in the beginning of solving $\mathscr{A}(\mathcal{X}) = \mathcal{G}_1$, we use again the F-orthonormal basis $\mathbb{V}_m$ and we extend it to

$\mathbb{V}_{m+1} = \{\mathbb{V}_m, \mathcal{V}_{new}\}$, where $\mathcal{V}_{new}$ is obtained by normalizing the residual as follows

$$\mathcal{V}_{new} = \frac{\mathcal{P}_1}{\|\mathcal{P}_1\|_F} \quad with \quad \mathcal{P}_1 = \mathscr{A}(\mathcal{X}_1) - \mathcal{G}_1.$$

We can then continue with $\mathscr{A}(\mathcal{X}) = \mathcal{G}_k$, for all $k = 2, \ldots$ in a similar manner. Thus, at each iteration k, we generate the following new block $\mathcal{V}_{new}$,

$$\mathcal{V}_{new} = \frac{\mathcal{P}_k}{\|\mathcal{P}_k\|_F} \quad with \quad \mathcal{P}_k = \mathscr{A}(\mathcal{X}_k) - \mathcal{G}_k, \tag{6.21}$$

that has to be added to the generalized tensor Krylov subspace basis $\mathbb{V}_{m+k-1}$.

On the other hand, finding the solution of the linear equation $\mathscr{A}(\mathcal{X}) = \mathcal{G}_k$ is equivalent to solving

$$\min_{\mathcal{X} \in \mathcal{X}_k + \mathcal{K}(\mathscr{A}, \mathcal{P}_k)} \|\mathcal{G}_k - \mathscr{A}(\mathcal{X})\|_F,$$

which can be written as

$$\min_{\mathcal{X} \in \mathcal{K}(\mathscr{A}, \mathcal{P}_k)} \|\mathcal{P}_k - \mathscr{A}(\mathcal{X})\|_F. \tag{6.22}$$

As a consequence, the approximate solution of (6.22) is given by

$$\mathcal{X}_{k+1} = \mathbb{V}_{m+k} \bar{\times}_{N+1} y, \tag{6.23}$$

since $\mathbb{V}_{m+k}$ is considered the basis of the generalized tensor Krylov subspace $\mathcal{K}(\mathscr{A}, \mathcal{P}_k)$. Working with this generalized Krylov subspace has the disadvantage that we lose the well-known relation that links the operation with the calculated basis and makes calculating y easier. Then, in order to simplify this minimization problem, we solve the linear problem

$$\mathscr{A}(\mathcal{X}) = \mathcal{P}_k,$$

which is equivalent to solving

$$\boldsymbol{vec}(\mathcal{P}_k) = [\boldsymbol{vec}(\mathscr{A}(\mathcal{V}_1)), \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_2)), \ldots, \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_{m+k})), \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_{new}))]y. \tag{6.24}$$

An optimized QR factorization can be used to solve this iterative equation. Following the same concept, we will calculate the QR factorization of the matrix $[\boldsymbol{vec}(\mathscr{A}(\mathcal{V}_1)), \ldots, \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_m))]$ in the first iteration (equation (6.20)), and then we will complete it in the next iterations as follows

$$[\boldsymbol{vec}(\mathrm{A}(\mathrm{V}_1)), \ldots, \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_{m+k-1})), \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_{new}))] = \underbrace{[Q_{m+k-1}, q_{new}]}_{Q_{m+k}} \underbrace{\begin{pmatrix} R_{m+k-1} & r_{new} \\ 0 & \rho_{new} \end{pmatrix}}_{R_{m+k}}, \tag{6.25}$$

where the new elements q_{new}, r_{new} and ρ_{new} are updated by the following relations

$$\begin{cases} \bullet \quad r_{new} = Q_{m+k-1}^T \, \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_{new})), \\ \bullet \quad \tilde{q}_{new} = \boldsymbol{vec}(\mathscr{A}(\mathcal{V}_{new})) - Q_{m+k-1} r_{new}, \\ \bullet \quad \rho_{new} = \|\tilde{q}_{new}\|_2, \\ \bullet \quad q_{new} = \dfrac{\tilde{q}_{new}}{\rho_{new}}. \end{cases} \tag{6.26}$$

Algorithm 6.1 *TTV_2/L^1-GMRES algorithm*

Given $\mathcal{B}$, $\mathcal{X}_0 = 0$, $\mathcal{R}_0$, $\mathcal{Y}_0$, $\mathcal{V}_0$, $\mathcal{W}_0$, μ, β, ρ, tol.
Compute $\mathcal{Y}_1$ and $\mathcal{R}_1$ using the formula (6.8) and (6.11) respectively.
Generate an F-orthonormal basis $\mathbb{V}_m = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_m\}$, an upper Hessenberg matrix H_m and a QR factorization using Arnoldi process.
Compute $\mathcal{X}_1$ the solution of $\mathscr{A}(\mathcal{X}) = \mathcal{G}_0$.
Update $\mathcal{V}_1$ and $\mathcal{W}_1$ using (6.7).
while $\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F / \|\mathcal{X}_k\|_F \leqslant tol$ **do**
    Compute $\mathcal{Y}_{k+1}$ and $\mathcal{R}_{k+1}$ using the formula (6.8) and (6.11) respectively.
    Update the F-orthonormal basis $\mathbb{V}_{k+m}$ by adding the new tensor block $\mathcal{V}_{new}$ using (6.21),
    Update the QR using (6.26),
    Determine y_k as a solution of (6.24) and compute $\mathcal{X}_{k+1} = \mathbb{V}_{k+m} \bar{\times}_{N+1} y_k$.
    Update $\mathcal{V}_{k+1}$ and $\mathcal{W}_{k+1}$ using (6.7).
end while
return $\mathcal{X}$.

6.3.2 Gradient optimized algorithm with fixed stepsize parameter

We can also apply the gradient method generalized in the tensor format to solve (6.15), this method consists in constructing a sequence $(\mathcal{X}_k)_k$ defined by

$$\mathcal{X}_{k+1} = \mathcal{X}_k + \alpha \mathcal{D}_k, \ \forall k \geqslant 0,$$

where $\mathcal{D}_k$ is the direction of descent, and α is the fixed stepsize parameter. We pick the direction of descent to be the one in the opposite direction of the gradient vector, which means we can take $\mathcal{D}_k$ to be the residual $\mathcal{R}_k = \mathscr{A}(\mathcal{X}_k) - \mathcal{G}_k$ in each step. We refer to this algorithm as a fixed step size gradient algorithm, since it does not require a line search at each step to determine α, in other words, the same step size α is used at each step k. Clearly, the convergence of the algorithm depends on the choice of α, and we would not expect the algorithm to work for arbitrary value of α. The theorem discussing a

necessary and sufficient condition on α for convergence of the algorithm can be found in [34]. We adopt the following choice of α

$$\alpha = \frac{\|\mathcal{D}_0\|_F^2}{\langle \mathscr{A}(\mathcal{D}_0)|\mathcal{D}_0\rangle}.$$

The following algorithm summarizes the TTV_2/L^1 approach using the conjugate gradient based on the tensor format.

Algorithm 6.2 TTV_2/L^1-*CG*

Given $\mathcal{G}, \mathcal{X}_0, \mathcal{R}_0, \mathcal{Y}_0, \mathcal{V}_0, \mathcal{W}_0, \mu, \beta, \rho, tol$.
Compute $\mathcal{Y}_1$ and $\mathcal{R}_1$ using the formula (6.8) and (6.11) respectively.
Gradient step: $\mathcal{D}_0 = \mathcal{G}_0 - \mathscr{A}(\mathcal{X}_0)$
    $\alpha = \dfrac{\|\mathcal{D}_0\|^2}{\langle \mathscr{A}(\mathcal{D}_0)|\mathcal{D}_0\rangle}.$
Compute: $\mathcal{X}_1 = \mathcal{X}_0 + \alpha\mathcal{D}_0$.
Update $\mathcal{V}_1$ and $\mathcal{W}_1$ using (6.7).
while $\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F / \|\mathcal{X}_k\|_F \leqslant tol$ **do**
    Compute $\mathcal{Y}_{k+1}$ and $\mathcal{R}_{k+1}$ using the formula (6.8) and (6.11) respectively
    Gradient step: $\mathcal{D}_{k+1} = \mathcal{G}_k - \mathscr{A}(\mathcal{X}_k)$
    Compute: $\mathcal{X}_{k+1} = \mathcal{X}_k + \alpha\mathcal{D}_{k+1}$.
    Update $\mathcal{V}_{k+1}$ and $\mathcal{W}_{k+1}$ using (6.7).
end while
return $\mathcal{X}$.

6.3.3 LSQR-BTF optimized algorithm

The *LSQR* method is an efficient numerical method for finding the solution of a linear equation based on the bidiagonalization procedure of Golub and Kahan [58]. In [78], the authors have constructed the *LSQR* method in the tensor form (*LSQR-BTF*) to solve tensor linear equations for different tensor products. Therefore, we can use *LSQR-BTF* to solve $\mathcal{X}$ subproblem in two different ways.

On the one hand, similarly to the optimized *GMRES-BTF*, we will adopt the same procedure to reduce the computation of the bidiagonalization process. That means, the bidiagonalization process applied to the the first tensor equation $\mathcal{A}(\mathcal{X}) = \mathcal{G}_0$ generates two basis $\mathbb{P}_m$ and $\mathbb{Q}_m$, that will extended in the next iteration by adding new blocks $\mathcal{P}_{new}$ and $\mathcal{Q}_{new}$, respectively. Then, at the iteration k, the basis will be in the form

$$\begin{aligned}\mathbb{P}_{m+k} &= [\mathbb{P}_{m+k-1}, \mathcal{P}_{new}],\\ \mathbb{Q}_{m+k} &= [\mathbb{Q}_{m+k-1}, \mathcal{Q}_{new}],\end{aligned}$$

where the new blocks $\mathcal{P}_{new}$ and $\mathcal{Q}_{new}$ are defined in every step k by

$$\mathcal{P}_{new} = \frac{\mathscr{A}(\mathcal{X}_k) - \mathcal{G}_k}{\|\mathscr{A}(\mathcal{X}_k) - \mathcal{G}_k\|_F}, \tag{6.27}$$

$$\mathcal{Q}_{new} = \mathscr{A}^T(\mathcal{P}_{new}).^1 \tag{6.28}$$

Once we get the new generalized basis, Givens rotations will be used to compute the approximated solution as describe in the following algorithm.

Algorithm 6.3 *TTV_2/L^1-LSQR 01 algorithm*

Given $\mathcal{B}$, $\mathcal{X}_0$, $\mathcal{R}_0$, $\mathcal{Y}_0$, $\mathcal{V}_0$, $\mathcal{W}_0$, μ, β, ρ, tol.
while $\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F / \|\mathcal{X}_k\|_F \leqslant tol$ **do**
  Compute $\mathcal{Y}_{k+1}$ and $\mathcal{R}_{k+1}$ using the formula (6.8) and (6.11) respectively.
  LSQR-BTF step :
    $\alpha_k = \|\mathscr{A}(\mathcal{X}_k) - \mathcal{G}_k\|_F$, $\mathcal{P}_k = (\mathscr{A}(\mathcal{X}_k) - \mathcal{G}_k)/\alpha_k$,
    $\alpha_k = \|\mathscr{A}^T(\mathcal{P}_k)\|_F$, $\mathcal{Q}_k = \mathscr{A}^T(\mathcal{P}_k)/\alpha_k$,
    $\bar{\phi}_k = \lambda_k, \bar{\rho}_k = \alpha_k$,
    $\lambda_k = \|\mathscr{A}(\mathcal{Q}_k) - \alpha_k \mathcal{P}_k\|$,
    $\rho_k = \sqrt{\bar{\rho}_k^2 + \lambda_k^2}; \quad c_k = \frac{\bar{\rho}_k}{\rho_k}; \quad and \quad \phi_k = c_k \bar{\phi}_k$.
  Compute: $\mathcal{X}_{k+1} = \mathcal{X}_k + \frac{\phi_k}{\rho_k}\mathcal{Q}_k$.
  Update $\mathcal{V}_{k+1}$ and $\mathcal{W}_{k+1}$ using (6.7).
end while
return $\mathcal{X}$.

On the other hand, by exploiting the possibility of solving unsymmetric problems using *LSQR*, the $\mathcal{X}$ subproblem can be written as follow

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmin}} \frac{\beta}{2}\|\mathscr{H}(\mathcal{X}) - \mathcal{B} - \mathcal{R} + \frac{1}{\beta}\mathcal{V}_k\|_F^2 + \frac{\rho}{2}\|\nabla\mathcal{X} - \mathcal{Y} + \frac{1}{\rho}\mathcal{W}_k\|_F^2, \tag{6.29}$$

which is equivalent to

$$\min_{\mathcal{X}} \|\mathscr{M}(\mathcal{X}) - \mathcal{G}_k\|_F^2, \tag{6.30}$$

using the definition of the operator $\mathscr{M}$ as

$$\begin{aligned} \mathscr{M} : \mathbb{R}^{I_1 \times \cdots \times I_N} &\longrightarrow \mathbb{R}^{(N+1)\cdot I_1 \times \cdots \times I_N} \\ \mathcal{X} &\longrightarrow \mathscr{M}(\mathcal{X}) = \begin{bmatrix} \sqrt{\frac{\beta}{2}}\mathscr{H}(\mathcal{X}) \\ \sqrt{\frac{\rho}{2}}\nabla\mathcal{X} \end{bmatrix}, \end{aligned}$$

and its transpose is defined as

$$\begin{aligned} \mathscr{M}^T : \mathbb{R}^{(N+1)\cdot I_1\times\cdots\times I_N} &\longrightarrow \mathbb{R}^{I_1\times\cdots\times I_N} \\ \mathcal{Y} = (\mathcal{Y}_1,\ldots,\mathcal{Y}_{N+1}) &\longrightarrow \mathscr{M}^T(\mathcal{Y}) = \sqrt{\frac{\beta}{2}}\mathscr{H}^T(\mathcal{Y}_1) + \sqrt{\frac{\rho}{2}}\sum_{n=1}^{N}\nabla_n^T\mathcal{Y}_{n+1} \end{aligned}$$

and the second member is given by $\mathcal{G}_k = \begin{bmatrix} \sqrt{\frac{\beta}{2}}(\mathcal{B}+\mathcal{R}-\frac{1}{\beta}\mathcal{V}_k) \\ \frac{\rho}{2}(\mathcal{Y}-\frac{1}{\rho}\mathcal{W}_k) \end{bmatrix}$. Therefore, we can apply the *LSQR-BTF* in each step k, to solve the tensor equation $\mathscr{M}(\mathcal{X}) = \mathcal{G}_k$.

Algorithm 6.4 *TTV_2/L^1-LSQR 02 algorithm*

1: Given $\mathcal{B}$, $\mathcal{X}_0$, $\mathcal{R}_0$, $\mathcal{Y}_0$, $\mathcal{V}_0$, $\mathcal{W}_0$, μ, β, ρ, tol.
2: **while** $\|\mathcal{X}_{k+1}-\mathcal{X}_k\|_F/\|\mathcal{X}_k\|_F \leqslant tol$ **do**
3: Compute $\mathcal{Y}_{k+1}$ and $\mathcal{R}_{k+1}$ using the formula (6.8) and (6.11) respectively.
4: Apply the *LSQR-BTF* algorithm to solve $\mathscr{M}(\mathcal{X}) = \mathcal{G}_k$.
5: Update $\mathcal{V}_{k+1}$ and $\mathcal{W}_{k+1}$ using (6.7).
6: **end while**
7: **return** $\mathcal{X}$.

6.4 Tensorial total variation TTV_1/L^2 problem

In this section, we consider solving the following TTV_1/L^2 problem

$$\min_{\mathcal{X}}\left(\mu\|\mathcal{X}-\mathcal{B}\|_F^2 + TTV_1(\mathcal{X})\right). \tag{6.31}$$

This minimization problem can be expressed according to Proposition 6.2.1 as

$$\min_{\mathcal{X}}\left(\mu\|\mathcal{X}-\mathcal{B}\|_F^2 + \sum_{n=1}^{N}|||\mathcal{X}\times_n C_n|||_1\right). \tag{6.32}$$

We first notice that the minimization problem (6.32) can be rewritten as the following equivalent constrained problem

$$\begin{cases} \min\limits_{\mathcal{X},\mathcal{Y}^{(1)},\ldots,\mathcal{Y}^{(N)}}\left(\mu\|\mathcal{X}-\mathcal{B}\|_F^2 + \sum\limits_{n=1}^{N}|||\mathcal{Y}^{(n)}|||_1\right) \\ \textit{subject to} \quad \mathcal{X}\times_n C_n = \mathcal{Y}^{(n)},\ \forall\ 1\leqslant n\leqslant N. \end{cases} \tag{6.33}$$

where the linear constraint of this problem can be written using the column block tensor form as $\nabla\mathcal{X} = \mathcal{Y}$, which is defined by

$$\mathcal{Y} = \begin{bmatrix} \mathcal{Y}^{(1)} \\ \vdots \\ \mathcal{Y}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathcal{X} \times_1 C_1 \\ \vdots \\ \mathcal{X} \times_N C_N \end{bmatrix} \in \mathbb{R}^{N \cdot I_1 \times \cdots \times I_N}. \tag{6.34}$$

The augmented Lagrangian function associate to (6.33) is defined as

$$\mathcal{L}_\beta(\mathcal{X}, \mathcal{Y}, \mathcal{W}) = \mu\|\mathcal{X} - \mathcal{B}\|_F^2 + \sum_{n=1}^{N} |||\mathcal{Y}^{(n)}|||_1 + \langle \mathcal{W} | \nabla\mathcal{X} - \mathcal{Y} \rangle + \frac{\beta}{2}\|\nabla\mathcal{X} - \mathcal{Y}\|_F^2,$$

where $\mathcal{W} \in \mathbb{R}^{N \cdot I_1 \times \cdots \times I_N}$ is the Lagrange multipliers of the linear constraint, and $\beta > 0$ is the penalty parameter for the violation of this linear constraint.

Since every norm is a convex function, by the triangle inequality and positive homogeneity, the objective function of the minimization problem (6.33) is convex as a sum of two convex functions. In addition, we minimize over linear constraints, which guarantees the convexity of the constraints. Therefore, the constrained problem (6.33) is convex because we minimize a convex function over convex constraints.

The classical alternating direction method of multipliers (*ADMM*) is used to solve this problem. The (*ADMM*) algorithm generates an iterative sequence $(\mathcal{X}_k, \mathcal{Y}_k, \mathcal{W}_k)_{k \geqslant 0}$ defined as follows: For given initial guesses $\mathcal{X}_0, \mathcal{Y}_0$ and $\mathcal{W}_0$, we have

$$\begin{cases} 1. \quad \mathcal{X}_{k+1} = \underset{\mathcal{X}}{\operatorname{argmin}}\, \mathcal{L}_\beta(\mathcal{X}, \mathcal{Y}_k, \mathcal{W}_k), \\ 2. \quad \mathcal{Y}_{k+1} = \underset{\mathcal{Y}}{\operatorname{argmin}}\, \mathcal{L}_\beta(\mathcal{X}_{k+1}, \mathcal{Y}, \mathcal{W}_k), \\ 3. \quad \mathcal{W}_{k+1} = \mathcal{W}_k + \beta(\nabla\mathcal{X}_{k+1} - \mathcal{Y}_{k+1}). \end{cases} \tag{6.35}$$

6.4.1 Solving $\mathcal{X}$-subproblem using LSQR-BTF

The $\mathcal{X}$-subproblem deduced from (6.35) is given by

$$\min_{\mathcal{X}} \left(\mu\|\mathcal{X} - \mathcal{B}\|_F^2 + \langle \mathcal{W}_k | \nabla\mathcal{X} - \mathcal{Y}_k \rangle + \frac{\beta}{2}\|\nabla\mathcal{X} - \mathcal{Y}_k\|_F^2 \right). \tag{6.36}$$

It is immediate to see that the minimization problem (6.36) is equivalent to:

$$\mathcal{X}_{k,\mu} = \underset{\mathcal{X}}{\operatorname{argmin}} \left\| \begin{bmatrix} \sqrt{\mu}(\mathcal{X} - \mathcal{B}) \\ \sqrt{\frac{\beta}{2}}(\nabla\mathcal{X} - \mathcal{Y}_k + \frac{1}{\beta}\mathcal{W}_k) \end{bmatrix} \right\|_F^2. \tag{6.37}$$

If we consider the variable changes $\mathcal{Z} = \mathcal{X} - \mathcal{B}$ and $\lambda = \frac{1}{\sqrt{\mu}}$, then the problem (6.37) becomes as follows

$$\mathcal{Z}_{k,\lambda} = \underset{\mathcal{Z}}{\operatorname{argmin}} \left\| \begin{bmatrix} \lambda^{-1}\mathcal{Z} \\ \sqrt{\frac{\beta}{2}}(\nabla\mathcal{Z} + \nabla\mathcal{B} - \mathcal{Y}_k + \frac{1}{\beta}\mathcal{W}_k) \end{bmatrix} \right\|_F^2. \tag{6.38}$$

In order to simplify this minimization problem, we consider the operator $\mathscr{M}$ defined as

$$\begin{array}{rcl} \mathscr{M} : \mathbb{R}^{I_1\times\cdots\times I_N} & \longrightarrow & \mathbb{R}^{N\cdot I_1\times\cdots\times I_N} \\ \mathcal{Z} & \longrightarrow & \mathscr{M}(\mathcal{Z}) = \sqrt{\frac{\beta}{2}}\nabla\mathcal{Z}. \end{array} \tag{6.39}$$

The transpose $\mathscr{M}^T$ of $\mathscr{M}$ is given by

$$\begin{array}{rcl} \mathcal{M}^T : \mathbb{R}^{N\cdot I_1\times\cdots\times I_N} & \longrightarrow & \mathbb{R}^{I_1\times\cdots\times I_N} \\ \mathcal{Y} = (\mathcal{Y}_1,\ldots,\mathcal{Y}_N) & \longrightarrow & \mathscr{M}^T(\mathcal{Y}) = \sqrt{\frac{\beta}{2}}\sum_{n=1}^{N}\nabla_n^T\mathcal{Y}_n \end{array}.$$

Let $\mathcal{G}_k = \sqrt{\frac{\beta}{2}}(\mathcal{Y}_k - \frac{1}{\beta}\mathcal{W}_k - \nabla\mathcal{B})$ denote the second member. The minimization problem (6.38) will be written in the following form

$$\mathcal{Z}_{k,\lambda} = \min_{\mathcal{Z}} \left\| \begin{bmatrix} \lambda^{-1}\mathcal{Z} \\ \mathscr{M}(\mathcal{Z}) - \mathcal{G}_k \end{bmatrix} \right\|_F^2. \tag{6.40}$$

As a result, $\mathcal{Z}_{k,\lambda}$ is the solution of the following normal equation associated with (6.40)

$$\mathscr{M}^T(\mathscr{M}(\mathcal{Z})) + \lambda^{-2}\mathcal{Z} = \mathscr{M}^T(\mathcal{G}_k). \tag{6.41}$$

We will update *LSQR-BTF* method [78, 91] to our tensor equation to determine an approximate solution $\mathcal{Z}_{k,\lambda}$. As an iterative method, TTV_1/L_2 problem requires solving $\mathcal{Y}$ and $\mathcal{X}$ subproblems at each iteration. Therefore, we need to solve the problem (6.41) in each step k. Instead of using full *LSQR-BTF* [78] method in every iteration, we propose to combine it with the *TTV* algorithm. First, the bidiagonalization process of Golub-Kahan [78, 58] applied to the operator $\mathscr{M}$ generates, at the first step ($k = 1$), the orthonormal tensor blocks $\{\mathcal{V}_1\}$ and $\{\mathcal{U}_1, \mathcal{U}_2\}$ satisfying

$$\left\{ \begin{array}{rcl} \mathscr{M}^T(\mathcal{U}_1) & = & \rho\mathcal{V}_1, \\ \mathscr{M}(\mathcal{V}_1) & = & \rho\mathcal{U}_1 + \sigma\mathcal{U}_2, \end{array} \right. \tag{6.42}$$

and

$$\rho = \|\mathscr{M}^T(\mathcal{U}_1)\|_F,\ \sigma = \|\mathscr{M}(\mathcal{V}_1) - \rho\mathcal{U}_1\|_F. \tag{6.43}$$

where the block $\mathcal{U}_1$ can be chosen as $\mathcal{G}_k/\|\mathcal{G}_k\|$. Then, by using only one step of *LSQR-BTF*,

the approximate solution $\mathcal{Z}_{k,\lambda}$ can be written in the form

$$\mathcal{Z}_{k,\lambda} = y_{k,\lambda}\mathcal{V}_1, \tag{6.44}$$

where $y_{k,\lambda}$ is a scalar. On the other hand, by using relations (6.42) and (6.44), the normal equation (6.41) leads to:

$$(\rho^2+\sigma^2)y_{k,\lambda} + \lambda^{-2}y_{k,\lambda} = \|\mathcal{G}_k\|_F\rho, \tag{6.45}$$

which simplifies the expression of the required scalar $y_{k,\lambda}$ as following:

$$y_{k,\lambda} = \frac{\rho\|\mathcal{G}_k\|_F}{\rho^2+\sigma^2+\lambda^{-2}}. \tag{6.46}$$

6.4.2 Regularization parameter selection

As we have seen in the previous subsection, the computation of an accurate approximation $\mathcal{Z}$ requires that a suitable value of the regularization parameter λ should be used. Let us first consider the sequence $(\lambda_k)_k$ of the regularization parameter λ_k, which mean that in every iteration k, we have an associated regularization parameter λ_k to our minimization problem. In fact, the idea behind using a sequence of parameters instead of a fixed value is to improve the convergence in practice, as well as make the performance less dependent on the initial choice of the parameter. To determine the parameter regularization λ_k, we will use the **discrepancy principle** approach [36]. We recall that the discrepancy principle method consists of finding a parameter λ_k satisfying the following equation

$$\|\mathcal{G}_k - \mathscr{M}(\mathcal{Z}_{k,\lambda_k})\|_F = \eta\delta,$$

where $\eta > 1$ is some predetermined real number and δ denotes an estimation of the norm of the error (see [36]). Then, the iterate $\mathcal{Z}_{k,\lambda_k}$ is the solution of the problem (6.40) corresponding to $\lambda = \lambda_k$, namely

$$\mathcal{Z}_{k,\lambda_k} = \underset{\mathcal{Z}}{\operatorname{argmin}} \left\| \begin{array}{c} \lambda_k^{-1}\mathcal{Z} \\ \mathscr{M}(\mathcal{Z}) - \mathcal{G}_k \end{array} \right\|_F^2. \tag{6.47}$$

To simplify the notation, we will denote $\mathcal{Z}_{k+1} = \mathcal{Z}_{k,\lambda_k}$, and then, the solution of the $\mathcal{X}$-subproblem at the step k is $\mathcal{X}_{k+1} = \mathcal{Z}_{k+1} + \mathcal{B}$.

For a fixed k, we consider the function defined for all $\lambda > 0$ by

$$\phi(\lambda) = \|\mathcal{G}_k - \mathscr{M}(\mathcal{Z}_{k,\lambda})\|_F^2. \tag{6.48}$$

The following result simplifies the expression of ϕ.

Proposition 6.4.1. *The expression $\phi(\lambda)$ of the function ϕ can be given in the following form,*

for all $\lambda > 0$*:*

$$\phi(\lambda) = \|\mathcal{G}_k\|_F^2 e_1^T \left(\lambda^2 \begin{pmatrix} \rho \\ \sigma \end{pmatrix} \begin{pmatrix} \rho & \sigma \end{pmatrix} + I_2\right)^{-2} e_1, \tag{6.49}$$

where the scalars ρ *and* σ *came from the relation* (6.42)*, and* e_1 *is the first unit vector of* $\mathbb{R}^2$*.*

Proof. Let $\mathcal{U}_2$ be the $N+1$ order tensor of size $I_1 \times \cdots \times I_N \times 2$ such that

$$\mathcal{U}_2(:,\ldots,:,i) = \mathcal{U}_i, \ \forall i = 1,2.$$

By the definition of n-mode (vector) product, and using the relation (6.42), we have

$$\phi(\lambda) = \left\| \mathcal{U}_2 \bar{\times}_{N+1} (\|\mathcal{G}_k\|_F e_1) - \mathcal{U}_2 \bar{\times}_{N+1} \begin{pmatrix} \rho \\ \sigma \end{pmatrix} y_{k,\lambda} \right\|_F^2 \tag{6.50}$$

where $y_{k,\lambda}$ is given by (6.46). As a consequence, we have

$$\begin{aligned}
\phi(\lambda) &= \left\| \|\mathcal{G}_k\|_F e_1 - \begin{pmatrix} \rho \\ \sigma \end{pmatrix} y_{k,\lambda} \right\|_2^2 \\
&= \left\| \|\mathcal{G}_k\|_F e_1 - \frac{\rho \|\mathcal{G}_k\|_F}{\rho^2 + \sigma^2 + \lambda^{-2}} \begin{pmatrix} \rho \\ \sigma \end{pmatrix} \right\|_2^2 \\
&= \|\mathcal{G}_k\|_F^2 \left\| \frac{1}{\lambda^2(\rho^2+\sigma^2)+1} \begin{pmatrix} \lambda^2\sigma^2 + 1 \\ -\lambda^2 \sigma \rho \end{pmatrix} \right\|_2^2 \\
&= \|\mathcal{G}_k\|_F^2 \left\| \left(\lambda^2 \begin{pmatrix} \rho \\ \sigma \end{pmatrix} \begin{pmatrix} \rho & \sigma \end{pmatrix} + I_2\right)^{-1} e_1 \right\|_2^2 \\
&= \|\mathcal{G}_k\|_F^2 e_1^T \left(\lambda^2 \begin{pmatrix} \rho \\ \sigma \end{pmatrix} \begin{pmatrix} \rho & \sigma \end{pmatrix} + I_2\right)^{-2} e_1.
\end{aligned}$$

□

Finally, to solve the nonlinear equation

$$\phi(\lambda) = \eta^2 \delta^2, \ \textit{for all } \lambda > 0, \tag{6.51}$$

a zero-finding method, such as Newton's method [41, 151], is used to find a suitable regularization parameter.

Remark 6.4.1. *Note that the function* $\mu \longrightarrow \phi(\mu)$*, which corresponds to the term* $\mu\|\mathcal{X} - \mathcal{B}\|_F^2$ *in* (6.36)*, is not guaranteed to be convex. Therefore, Newton method has to be safeguarded when applied to the solution of* $\phi(1/\sqrt{\mu}) = \eta\delta$*. It is the reason for considering* $\lambda = \frac{1}{\sqrt{\mu}}$ *[12].*

6.4.3 Solving $\mathcal{Y}$-subproblem

According to (6.35), the $\mathcal{Y}$-subproblem represents as the following unconstrained minimization problem

$$\min_{\mathcal{Y}}\left(\sum_{n=1}^{N}|||\mathcal{Y}^{(n)}|||_1+\frac{\beta}{2}\|\nabla\mathcal{X}_{k+1}-\mathcal{Y}\|_F^2+\langle\mathcal{W}_k|\nabla\mathcal{X}_{k+1}-\mathcal{Y}\rangle\right), \tag{6.52}$$

that will be solved by applying the shrinkage formula [106] as proved in the following result.

Proposition 6.4.2. *Given $\mathcal{X}_{k+1}$, computing the iterate $\mathcal{Y}_{k+1}$ from the problem (6.52) is equivalent to compute*

$$\mathcal{Y}^{(n)}=\max\left\{\mathcal{K}^{(n)}-\frac{1}{\beta},0\right\}sign(\mathcal{K}^{(n)}),\ \forall\ 1\leqslant n\leqslant N \tag{6.53}$$

where $\mathcal{K}^{(n)}=\nabla_{(n)}\mathcal{X}_{k+1}+\frac{1}{\beta}\mathcal{W}_k^{(n)}$.

Proof. Note that, the solution $\mathcal{Y}_{k+1}$ can be obtained by solving the problem

$$\begin{aligned}\mathcal{Y}_{k+1}&=\underset{\mathcal{Y}}{\operatorname{argmin}}\left(\sum_{n=1}^{N}|||\mathcal{Y}^{(n)}|||_1+\frac{\beta}{2}\|\nabla\mathcal{X}_{k+1}-\mathcal{Y}\|_F^2+\langle\mathcal{W}_k|\nabla\mathcal{X}_{k+1}-\mathcal{Y}\rangle\right),\\&=\underset{\mathcal{Y}^{(1)},\ldots,\mathcal{Y}^{(N)}}{\operatorname{argmin}}\left(\sum_{i_1,\ldots,i_N}\sum_{n=1}^{N}|\mathcal{Y}^{(n)}_{i_1,\ldots,i_N}|+\frac{\beta}{2}\left\|\mathcal{Y}-\nabla\mathcal{X}_{k+1}-\frac{1}{\beta}\mathcal{W}_k\right\|_F^2\right),\\&=\underset{\mathcal{Y}^{(1)},\ldots,\mathcal{Y}^{(N)}}{\operatorname{argmin}}\left(\sum_{i_1,\ldots,i_N}\sum_{n=1}^{N}|\mathcal{Y}^{(n)}_{i_1,\ldots,i_N}|+\frac{\beta}{2}\sum_{n=1}^{N}\left\|\mathcal{Y}^{(n)}-\nabla_{(n)}\mathcal{X}_{k+1}-\frac{1}{\beta}\mathcal{W}_k^{(n)}\right\|_F^2\right),\\&=\underset{\mathcal{Y}^{(1)},\ldots,\mathcal{Y}^{(N)}}{\operatorname{argmin}}\left[\sum_{i_1,\ldots,i_N}\left(\sum_{n=1}^{N}|\mathcal{Y}^{(n)}_{i_1,\ldots,i_N}|+\frac{\beta}{2}\sum_{n=1}^{N}(\mathcal{Y}^{(n)}_{i_1,\ldots,i_N}-\mathcal{K}^{(n)}_{i_1,\ldots,i_N})^2\right)\right],\end{aligned} \tag{6.54}$$

where $\mathcal{K}^{(n)}=\nabla_{(n)}\mathcal{X}_{k+1}+\frac{1}{\beta}\mathcal{W}_k^{(n)}$. We remark that the minimization subproblem (6.54) is separable, then it is equivalent to

$$\min_{\mathcal{Y}_{i_1,\ldots,i_N}}\sum_{n=1}^{N}|\mathcal{Y}^{(n)}_{i_1,\ldots,i_N}|+\frac{\beta}{2}\sum_{n=1}^{N}(\mathcal{Y}^{(n)}_{i_1,\ldots,i_N}-\mathcal{K}^{(n)}_{i_1,\ldots,i_N})^2,\ \forall i_1,\ldots,i_N$$

for which the unique minimizer is given by the following well-known one dimensional shrinkage formula [106]:

$$\mathcal{Y}^{(n)}_{i_1,\ldots,i_N}=\max\left\{\mathcal{K}^{(n)}_{i_1,\ldots,i_N}-\frac{1}{\beta},0\right\}sign(\mathcal{K}^{(n)}_{i_1,\ldots,i_N})\quad\forall n,\quad\forall i_1,\ldots,i_N, \tag{6.55}$$

which establishes the formula. □

6.4.4 Penalty parameter selection

An optimal choice of the penalty parameter plays a crucial role in obtaining good convergence. A standard extension is to use different penalty parameters β_k for each iteration. To define the value of β_k at each iteration k, we need to define the *ADMM* residuals of the problem (6.33). Let $\mathcal{R}_k$ and $\mathcal{S}_k$ denote the primal residual and the dual residual associate to the primal problem (6.33) and its dual problem, respectively,

$$\begin{aligned}\mathcal{R}_{k+1} &= \nabla\mathcal{X}_{k+1} - \mathcal{Y}_{k+1},\\ \mathcal{S}_{k+1} &= \beta_k(\mathcal{Y}_{k+1} - \mathcal{Y}_k).\end{aligned}$$

These two residuals converge to zero as *ADMM* proceeds [22]. In addition, the norm of the primal residual decreases with increasing β_k. On the other hand, the definition of the dual residual suggests that it increases with increasing β_k . For that reason, in [22, 147], they proposed the following simple scheme to choose β_k

$$\beta_{k+1} = \begin{cases} \tau\beta_k & if \quad \|\mathcal{R}_k\|_F > \nu\|\mathcal{S}_k\|_F, \\ \tau^{-1}\beta_k & if \quad \|\mathcal{S}_k\|_F > \nu\|\mathcal{R}_k\|_F, \\ \beta_k & otherwise, \end{cases} \tag{6.56}$$

where $\tau > 1$ and $\nu > 1$ are constants. The usual values are $\tau = 2$ and $\nu = 10$ in [75, 22, 147]. The idea behind this penalty parameter update is trying to keep the primal and dual residual norms within a factor of ν of one another as they both converge to zero.

The following algorithm summarizes the main steps to compute a regularization parameter, the penalty parameter, and the corresponding regularized solution of (6.31).

6.5 Existence of the solution and convergence of the TTV algorithm

Theorem 6.5.1. *If $(\mathcal{X}^*, \mathcal{R}^*, \mathcal{Y}^*, \mathcal{V}^*, \mathcal{W}^*)$ is a saddle point of $\mathscr{L}_{\beta,\rho}$, then $\mathcal{X}^*$ is a solution of (6.6).*

Proof. Let $(\mathcal{X}^*, \mathcal{R}^*, \mathcal{Y}^*, \mathcal{V}^*, \mathcal{W}^*)$ be a saddle point of $\mathscr{L}_{\beta,\rho}$. By the definition of a saddle point, we have, for all $(\mathcal{X}, \mathcal{R}, \mathcal{Y}, \mathcal{V}, \mathcal{W})$,

$$\mathscr{L}_{\beta,\rho}(\mathcal{X}^*, \mathcal{R}^*, \mathcal{Y}^*, \mathcal{V}, \mathcal{W}) \leq \mathscr{L}_{\beta,\rho}(\mathcal{X}^*, \mathcal{R}^*, \mathcal{Y}^*, \mathcal{V}^*, \mathcal{W}^*) \leq \mathscr{L}_{\beta,\rho}(\mathcal{X}, \mathcal{R}, \mathcal{Y}, \mathcal{V}^*, \mathcal{W}^*). \tag{6.57}$$

Using the first part of (6.57), we got

$$\langle \mathcal{V} - \mathcal{V}^* | \mathscr{H}(\mathcal{X}^*) - \mathcal{B} - \mathcal{R}^* \rangle + \langle \mathcal{W} - \mathcal{W}^* | \nabla\mathcal{X}^* - \mathcal{Y}^* \rangle \leqslant 0,$$

Algorithm 6.5 *TTV_1/L^2-LSQR algorithm*

Given $\mathcal{B}$, $\mathcal{X}_0 = \mathbf{0}$, $\mathcal{Y}_0$, $\mathcal{W}_0$, β_0, δ, tol.
while $\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F / \|\mathcal{X}_k\|_F \leqslant tol$ **do**
  LSQR-BTF step : $\alpha_k = \|\mathcal{G}_k\|_F, \quad \mathcal{U}_k = \mathcal{G}_k/\alpha_k,$
  $\sigma_k = \|\mathscr{M}^T(\mathcal{U}_k)\|_F, \quad \mathcal{V}_k = \mathscr{M}^T(\mathcal{U}_k)/\sigma_k,$
  $\rho_k = \|\mathscr{M}(\mathcal{V}_k) - \sigma_k \mathcal{U}_k\|_F.$
  Determine the regularization parameter μ_k by using Newton's method to solve (6.51).
  Compute $\mathcal{Z}_{k,\lambda_k} = y_{k,\lambda_k}\mathcal{V}_k$ where $y_{k,\lambda_k} = \dfrac{\|\mathcal{G}_k\|_F \rho_k}{\rho_k^2 + \sigma_k^2 + \lambda_k^{-2}}$.
  Compute $\mathcal{X}_{k+1} = \mathcal{Z}_{k,\lambda_k} + \mathcal{B}$.
  Determine the penalty parameter β_{k+1} using the formula (6.56).
  Compute $\mathcal{Y}_{k+1}$ using the formula (6.53).
  Update the Lagrange multiplier $\mathcal{W}_{k+1}$ using (6.35).
end while
return $\mathcal{X}$.

since this holds for every $(\mathcal{V}, \mathcal{W})$, we obtain $\mathscr{H}(\mathcal{X}^*) - \mathcal{B} - \mathcal{R}^* = 0$ and $\nabla\mathcal{X}^* - \mathcal{Y}^* = 0$. In the other hand, choosing $\mathcal{R} = \mathscr{H}(\mathcal{X}) - \mathcal{B}$ and $\mathcal{Y} = \nabla\mathcal{X}$ in the second part of (6.57) leads to

$$\forall \mathcal{X}, \quad \mathscr{F}(\mathscr{H}(\mathcal{X}^*) - \mathcal{B}) + \mathscr{G}(\nabla\mathcal{X}^*) \leqslant \mathscr{F}(\mathscr{H}(\mathcal{X}) - \mathcal{B}) + \mathscr{G}(\nabla\mathcal{X}),$$

which is clearly equivalent to the following inequality:

$$|||\mathscr{H}(\mathcal{X}^*) - \mathcal{B}|||_1 + \mu TTV_2(\mathcal{X}^*) \leqslant |||\mathscr{H}(\mathcal{X}) - \mathcal{B}|||_1 + \mu TTV_2(\mathcal{X}), \ \forall \mathcal{X}.$$

As a result, we deduce that $\mathcal{X}^*$ is a solution of (6.6).
For the inverse implication, we refer the reader to [141, 148]. □

Theorem 6.5.2. *A point $(\mathcal{X}^*, \mathcal{R}^*, \mathcal{Y}^*, \mathcal{V}^*, \mathcal{W}^*)$ is a saddle point of $\mathscr{L}_{0,0}$ if and only if it is a saddle point of $\mathscr{L}_{\beta,\rho}$, for all $\beta, \rho \geqslant 0$. Note that $\mathscr{L}_{0,0}$ is exactly the Lagrangian functional $\mathscr{L}$ associated to (6.6).*

Proof. For the proof we refer the reader to [22]. □

Theorem 6.5.3. *Assume that $(\mathcal{X}^*, \mathcal{R}^*, \mathcal{Y}^*, \mathcal{V}^*, \mathcal{W}^*)$ is a saddle point of $\mathscr{L}_{\beta,\rho}$. The sequence $(\mathcal{X}_k, \mathcal{R}_k, \mathcal{Y}_k, \mathcal{V}_k, \mathcal{W}_k)$ generated by TTV_2 Algorithms (6.1, 6.2, 6.4) satisfies :*

1. $\lim\limits_{k \to +\infty} (\mathscr{F}(\mathcal{R}_k) + \mathscr{G}(\mathcal{Y}_k)) = \mathscr{F}(\mathcal{R}^*) + \mathscr{G}(\mathcal{Y}^*),$
2. $\lim\limits_{k \to +\infty} \|\mathscr{H}(\mathcal{X}_k) - \mathcal{B} - \mathcal{R}_k\|_F^2 = 0,$
3. $\lim\limits_{k \to +\infty} \|\nabla\mathcal{X}_k - \mathcal{Y}_k\|_F^2 = 0.$

Proof. We begin by replacing the couple $(\mathcal{V},\mathcal{W})$ by $(\mathscr{H}(\mathcal{X}^*)-\mathcal{B}-\mathcal{R}^*,\mathcal{W}^*)$ in the first inequality of (6.57), we obtain

$$\mathscr{H}(\mathcal{X}^*)-\mathcal{B}-\mathcal{R}^*=0. \tag{6.58}$$

In the other hand, if we take the couple $(\mathcal{V},\mathcal{W})$ to be $(\mathcal{V}^*,\nabla\mathcal{X}^*-\mathcal{Y}^*)$, we get

$$\nabla\mathcal{X}^*-\mathcal{Y}^*=0. \tag{6.59}$$

The main idea of the proof is to increase the three terms by quantities that converge to zero which will guaranty the convergence of the three expressions. According to the theorem 6.5.2, $(\mathcal{X}^*,\mathcal{R}^*,\mathcal{Y}^*,\mathcal{V}^*,\mathcal{W}^*)$ is a saddle point of $\mathscr{L}_{0,0}$, which means

$$\forall(\mathcal{X},\mathcal{R},\mathcal{Y}),\quad \mathscr{L}_{0,0}(\mathcal{X}^*,\mathcal{R}^*,\mathcal{Y}^*,\mathcal{V}^*,\mathcal{W}^*)\leq\mathscr{L}_{0,0}(\mathcal{X},\mathcal{R},\mathcal{Y},\mathcal{V}^*,\mathcal{W}^*).$$

If we take $(\mathcal{X},\mathcal{R},\mathcal{Y})$ to be $(\mathcal{X}_k,\mathcal{R}_k,\mathcal{Y}_k)$, we get

$$\mathscr{F}(\mathcal{R}^*)+\mathscr{G}(\mathcal{Y}^*)-\mathscr{F}(\mathcal{R}_k)-\mathscr{G}(\mathcal{Y}_k)\leq\langle\mathcal{V}^*|\mathscr{H}(\hat{\mathcal{X}}_k)-\hat{\mathcal{R}}_k\rangle+\langle\mathcal{W}^*|\nabla\hat{\mathcal{X}}_k-\hat{\mathcal{Y}}_k\rangle, \tag{6.60}$$

where $\hat{\mathcal{X}}_k=\mathcal{X}_k-\mathcal{X}^*$, $\hat{\mathcal{Y}}_k=\mathcal{Y}_k-\mathcal{Y}^*$, $\hat{\mathcal{R}}_k=\mathcal{R}_k-\mathcal{R}^*$, $\hat{\mathcal{V}}_k=\mathcal{V}_k-\mathcal{V}^*$, and $\hat{\mathcal{W}}_k=\mathcal{W}_k-\mathcal{W}^*$. Also $(\mathcal{X}_k,\mathcal{R}_k,\mathcal{Y}_k,\mathcal{V}_k,\mathcal{W}_k)$ is a saddle point of $\mathscr{L}_{\beta,\rho}$, as a consequence, it is a saddle point of $\mathscr{L}_{2\beta,2\rho}$, which give us the following inequality using (6.58) and (6.59).

$$\begin{aligned}\mathscr{F}(\mathcal{R}^*)+\mathscr{G}(\mathcal{Y}^*)-\mathscr{F}(\mathcal{R}_k)-\mathscr{G}(\mathcal{Y}_k)\geqslant{}&\langle\mathcal{V}_k|\mathscr{H}(\hat{\mathcal{X}}_k)-\hat{\mathcal{R}}_k\rangle+\langle\mathcal{W}_k|\nabla\hat{\mathcal{X}}_k-\hat{\mathcal{Y}}_k\rangle\\&+\beta\|\mathscr{H}(\mathcal{X}_k)-\mathcal{B}-\mathcal{R}_k\|_F^2+\rho\|\nabla\mathcal{X}_k-\mathcal{Y}_k\|_F^2.\end{aligned} \tag{6.61}$$

From (6.60) and (6.61), we deduce that

$$\beta\|\mathscr{H}(\mathcal{X}_k)-\mathcal{B}-\mathcal{R}_k\|_F^2+\rho\|\nabla\mathcal{X}_k-\mathcal{Y}_k\|_F^2\leq-\langle\hat{\mathcal{V}}_k|\mathscr{H}(\hat{\mathcal{X}}_k)-\hat{\mathcal{R}}_k\rangle-\langle\hat{\mathcal{W}}_k|\nabla\hat{\mathcal{X}}_k-\hat{\mathcal{Y}}_k\rangle. \tag{6.62}$$

Considering now the sequence $(u_k)_{k\geqslant0}$ defined as

$$u_k=\rho\|\hat{\mathcal{V}}_k\|+\beta\|\hat{\mathcal{W}}_k\|,$$

this sequence decreases according to (6.62) since

$$\begin{aligned}u_k-u_{k+1}={}&-\beta^2\rho\|\mathscr{H}(\mathcal{X}_k)-\mathcal{B}-\mathcal{R}_k\|_F^2-\rho^2\beta\|\nabla\mathcal{X}_k-\mathcal{Y}_k\|_F^2-2\rho\beta\langle\hat{\mathcal{V}}_k|\mathscr{H}(\hat{\mathcal{X}}_k)-\hat{\mathcal{R}}_k\rangle\\&-2\rho\beta\langle\hat{\mathcal{W}}_k|\nabla\hat{\mathcal{X}}_k-\hat{\mathcal{Y}}_k\rangle.\end{aligned}$$

As a result, we have

$$\beta^2\rho\|\mathscr{H}(\mathcal{X}_k)-\mathcal{B}-\mathcal{R}_k\|_F^2+\beta\rho^2\|\nabla\mathcal{X}_k-\mathcal{Y}_k\|_F^2\leqslant u_k-u_{k+1}. \tag{6.63}$$

By regrouping terms, we obtain

$$\sum_{k=0}^{\infty} \beta^2 \rho \|\mathscr{H}(\mathcal{X}_k) - \mathcal{B} - \mathcal{R}_k\|_F^2 + \rho^2 \beta \|\nabla \mathcal{X}_k - \mathcal{Y}_k\|_F^2 \leqslant u_0. \tag{6.64}$$

So that $\lim_{k\to+\infty} \|\mathscr{H}(\mathcal{X}_k) - \mathcal{B} - \mathcal{R}_k\|_F^2 = 0$, and $\lim_{k\to+\infty} \|\nabla \mathcal{X}_k - \mathcal{Y}_k\|_F^2 = 0$.

From (6.60) and (6.61), we obtain

$$\mathscr{F}(\mathcal{R}^*) + \mathscr{G}(\mathcal{Y}^*) - \mathscr{F}(\mathcal{R}_k) - \mathscr{G}(\mathcal{Y}_k) \leqslant \|\mathcal{V}^*\|_F \|\mathscr{H}(\hat{\mathcal{X}}_k) - \hat{\mathcal{R}}_k\|_F + \|\mathcal{W}^*\|_F \|\nabla \hat{\mathcal{X}}_k - \hat{\mathcal{Y}}_k\|_F.$$

and

$$\begin{aligned}\mathscr{F}(\mathcal{R}_k) + \mathscr{G}(\mathcal{Y}_k) - \mathscr{F}(\mathcal{R}^*) - \mathscr{G}(\mathcal{Y}^*) \leq \|\mathcal{V}_k\|_F \|\mathscr{H}(\hat{\mathcal{X}}_k) - \hat{\mathcal{R}}_k\|_F + \|\mathcal{W}_k\|_F \|\nabla \hat{\mathcal{X}}_k - \hat{\mathcal{Y}}_k\|_F \\ - \beta \|\mathscr{H}(\mathcal{X}_k) - \mathcal{B} - \mathcal{R}_k\|_F^2 - \rho \|\nabla \mathcal{X}_k - \mathcal{Y}_k\|_F^2.\end{aligned}$$

Passing to the limit in both equations as $k \longrightarrow \infty$, we obtain desired convergence of our algorithms.

□

Remark 6.5.1. *Note that the existence of the solution and convergence of the TTV_1/L^2 algorithm, can be proved using the same process.*

6.6 Subproblem Reduction using truncated *SVD*

The normal equation associate to the $\mathcal{X}$ subproblem (6.13) provides a linear Sylvester structure that will allow a possible reduction using Truncated Singular Value Decomposition (TSVD) [72] of the matrices H_1 and H_2. Let us denote the *TSVD* decomposition of the matrices H_1 and H_2 by

$$\begin{cases} H_1 = U_1 \Sigma_1 V_1^T, \\ H_2 = U_2 \Sigma_2 V_2^T, \end{cases} \tag{6.65}$$

where $U_1, V_1 \in \mathbb{R}^{I_1 \times r_1}$, $\Sigma_1 \in \mathbb{R}^{r_1 \times r_1}$ and $U_2, V_2 \in \mathbb{R}^{I_2 \times r_2}$, $\Sigma_2 \in \mathbb{R}^{r_2 \times r_2}$, with $r_1 \ll I_1$ and $r_2 \ll I_2$. The normal equation associated to (6.12) can be written also as:

$$\rho(\mathcal{X} \times_1 H_1^T H_1 \times_2 H_2^T H_2) + \beta \sum_{n=1}^{N} \mathcal{X} \times_n C_n^T C_n = \mathcal{G}_k. \tag{6.66}$$

On the other hand, by setting $\mathcal{Y} = \mathcal{X} \times_1 V_1^T \times_2 V_2^T$, the equation (6.66) will be expressed in the following form

$$\rho(\mathcal{Y} \times_1 \Sigma_1^2 \times_2 \Sigma_2^2) + \beta(\mathcal{Y} \times_1 V_1^T \tilde{C}_1 V_1 + \mathcal{Y} \times_2 V_2^T \tilde{C}_2 V_2 + \sum_{n=3}^{N} \mathcal{Y} \times_n \tilde{C}_n) = \mathcal{S}_k, \tag{6.67}$$

where $\mathcal{S}_k = \mathcal{G}_k \times_1 V_1^T \times_2 V_2^T \in \mathbb{R}^{r_1 \times r_2 \times I_3 \times \cdots \times I_N}$ and $\tilde{C}_i = C_i^T C_i$, *for* $i = 1, 2, 3$. As a result, we reduce the size of the tensor equation from $I_1 \times I_2 \times I_3 \times \cdots \times I_N$ into $r_1 \times r_2 \times I_3 \times \cdots \times I_N$.

In the literature, there is a class of methods developed to compute an optimal parameters of truncation r_1 and r_2 for matrix ill-posed problems. Among the well-known, we refer the reader to [17, 149]. In our case, the tensorial form of the linear problem (6.13) makes the computation of an appropriate truncation parameter more complex, therefore, in the numerical test, the parameters r_1 and r_2 are chosen empirically to yield the best reconstruction.

6.7 Conclusion

In this chapter, we have introduced the regularization term of the discrete tensor total variation. Two particular models have been studied using convex optimization. The main idea is based on dividing our minimization problems into feasible optimization subproblems. Projection techniques are developed in an optimized way to solve this subproblems and accelerate the convergence. On the other hand, the discrepancy principle is updated to compute an optimal regularization parameter which is complicated to compute in general in case of total variation regularization.

CHAPTER 7

Constrained tensorial total variation problem via like-gradient methods

7.1 Introduction

We consider solving a class of constrained tensor regularization problem of the form

$$\min_{\mathcal{X}\in\Omega}\left(\frac{1}{2}\|\mathscr{H}(\mathcal{X})-\mathcal{B}\|_F^2+\mu|||\nabla\mathcal{X}|||_1\right), \tag{7.1}$$

where the solution $\mathcal{X}$ and the observation $\mathcal{B}$ are Nth-order tensors, $\mathscr{H}$ is a given linear tensor operator and the set Ω is assumed to be a convex constraint over $\mathcal{X}$. The regularization term consists of the tensorial total variation regularization operator $|||\nabla\mathcal{X}|||_1$ and the positive regularization parameter μ. The proposed problem (7.1) represents a constrained multidimensional total variation regularization problem that will cover a wide range of application fields, such as color image and video processing.

7.2 An alternating tensorial conditional gradient method

Let us start by the particular case where the tensor operation $\mathscr{H}$ consider as the identity operator. In the current section, we describes a new iterative method to solve a convex multidimensional minimization problem of the form

$$\min_{\mathcal{X}\in\Omega}\left(\frac{\mu}{2}\|\mathcal{X}-\mathcal{G}\|^2+|||\nabla\mathcal{X}|||_1\right). \tag{7.2}$$

The continuous model and the structure of the regularization total variation term lead to a new developed approach that attempts to solve the convex optimization problem by breaking it into feasible subproblems. A key step of this method is the splitting of variables guaranteed by introducing the idea of the pseudo-Lagrangian operator followed by an alternating approach. The latter generates a sequence that converge to the solution of our optimization problem.

7.2.1 From the continuous to discrete model

In order to understand the definition of the discrete total variation in the tensorial case, let first examine the continuous form of the model (7.2) given as follows

$$\min_{f\in C}\left(TV(f)+\frac{\mu}{2}\|f-g\|_2^2\right), \tag{7.3}$$

where f, $g : \boldsymbol{E} \subset \mathbb{R}^N \to \mathbb{R}$ denote the desired original data and the contaminated data respectively, and the set C denotes a convex functional set in $BV(\boldsymbol{E})$. The space $BV(\boldsymbol{E})$ is defined as the space of functions of bounded variation (for more details about the space $BV(\boldsymbol{E})$ and its properties, we refer the readers to [2] and the references therein) and is given as follows:

$$BV(\boldsymbol{E}) = \left\{f \in L^1(\boldsymbol{E}),\ TV(f) < \infty\right\},$$

where $L^1(\boldsymbol{E})$ stands for the classical Lebesgue space defined on the domain $\boldsymbol{E}$. Let us consider the convex set K defined by

$$K = \left\{\phi \in \mathcal{C}_c^1(\boldsymbol{E},\mathbb{R}^N), \|\phi\|_\infty \leqslant 1\right\}, \tag{7.4}$$

where $\mathcal{C}_c^1(\boldsymbol{E},\mathbb{R}^N)$ denotes the space of functions from $\boldsymbol{E}$ to $\mathbb{R}^N$ of class $\mathcal{C}^1$ compactly supported on $\boldsymbol{E}$, the function $\phi : \boldsymbol{E} \longrightarrow \mathbb{R}^N$ is the dual function and f is the primal function. The norm $\|\cdot\|_\infty$ is given by

$$\|\omega\|_\infty = \sup_{x\in\boldsymbol{E}} |\omega(x)|, \quad \forall \omega \in \mathcal{C}_c^1(\boldsymbol{E},\mathbb{R}^N),$$

with $|\omega(x)|$ stands for the Euclidean norm of $\omega(x)$ in $\mathbb{R}^N$. Then, the total variation semi-norm, denotes also TV term, is given by

$$TV(f) = \sup_{\phi\in K}\langle \nabla f, \phi\rangle, \tag{7.5}$$

where the dual product $\langle .,. \rangle$ is given by

$$\langle \nabla f, \phi\rangle = \int_{\boldsymbol{E}} \nabla f(x)\cdot\phi(x)dx,$$

where the notation $x \cdot y$ stands for the standard scalar product of two vectors x and y in $\mathbb{R}^N$, and the gradient of f is denoted by ∇f. In a particular case when f is smooth, by using Cauchy-Schwartz inequality [25], the TV term will be given as:

$$TV(f) = \int_{\boldsymbol{E}} |\nabla f(x)| dx. \tag{7.6}$$

Recall that the main advantage of considering the total variation as a regularization operator instead of other operators like Tikonov's [19, 26, 59], is the ability of preserving edges. The TV term discourages the solution of the problem from having oscillations. Moreover, it allows the solution to have discontinuities, which is the case of real modeled problems. According to [67], if $g \in L^2(\boldsymbol{E})$, the minimizer of (7.3) exists, and it is unique and stable in $L^2(\boldsymbol{E})$ with respect the perturbations of f. The drawback of this method is in the choice of the regularization parameter which will be complicated because of the TV term structure. The generalized cross-validation (GCV) [60], Morozov discrepancy principle [36], and the L-curve method [69] are among the most efficient methods to compute an optimal regularization parameter. However, it is difficult to adapt such techniques in the case of total variation because of the nonlinearity of the model.

Now, let us back to the discrete formulation of our main problem, when all results are developed in the finite-multidimensional setting. Let the tensors $\mathcal{X}, \mathcal{W}$ and $\mathcal{G}$ represent the discrete form corresponding to the functions f, ϕ and g, respectively. Let Ω and $\mathcal{K}$ denote the convex sets in the discrete tensor version of C and K, respectively. For the simplicity of the notation, the space of Nth-order tensor $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ will be denoted by $\mathbb{T}_N$ and the set $\mathbb{T}_N^N := \mathbb{T}_N \times \cdots \times \mathbb{T}_N$ will be denoted by $\mathbb{S}_N$.

Definition 7.2.1. *The discrete divergence operator* $\operatorname{div} : \mathbb{S}_N \longrightarrow \mathbb{T}_N$ *is defined such that, for any* $\mathcal{Y} \in \mathbb{S}_N$ *and* $\mathcal{X} \in \mathbb{T}_N$ *, we have*

$$\langle -\operatorname{div}(\mathcal{Y}) \,|\, \mathcal{X} \rangle = \langle \mathcal{Y} \,|\, \nabla \mathcal{X} \rangle. \tag{7.7}$$

Let us consider the Nth-order tensor operator $\partial^{(n)} : \mathbb{S}_N \longrightarrow \mathbb{T}_N$ defined as

$$(\partial^{(n)}(\mathcal{Y}))_{i_1,\ldots,i_n,\ldots,i_N} = \begin{cases} (\mathcal{Y}^{(n)})_{i_1,\ldots,i_n,\ldots,i_N} - (\mathcal{Y}^{(n)})_{i_1,\ldots,i_n-1,\ldots,i_N}, & \text{if } 1 < i_n < I_n, \\ (\mathcal{Y}^{(n)})_{i_1,\ldots,i_N}, & \text{if } i_n = 1, \\ -(\mathcal{Y}^{(n)})_{i_1,\ldots,i_n-1,\ldots,i_N}, & \text{if } i_n = I_n, \end{cases} \tag{7.8}$$

Then, the discrete divergence $\operatorname{div}(\mathcal{Y})$ of an element $\mathcal{Y} = (\mathcal{Y}^{(1)}, \ldots, \mathcal{Y}^{(N)})$ in $\mathbb{S}_N$ is an element of $\mathbb{T}_N$ defined by

$$\operatorname{div}(\mathcal{Y}) = \sum_{n=1}^{N} \partial^{(n)}(\mathcal{Y}). \tag{7.9}$$

Lemma 7.2.1. *For all* $\mathcal{Y} \in \mathbb{S}_N$*, we have the following inequality*

$$\|\operatorname{div}(\mathcal{Y})\|_F^2 \leqslant 2^{N+1} \|\mathcal{Y}\|_F^2. \tag{7.10}$$

Proof. Based on the fact that $(a_1 + a_2)^2 \leqslant 2(a_1^2 + a_2^2)$, for all a_1 and a_2 in $\mathbb{R}$, we may show, by induction on N, that for all $a_i \in \mathbb{R}$,

$$\left(\sum_{i=1}^{N} a_i\right)^2 \leqslant 2^{N-1} \sum_{i=1}^{N} a_i^2. \tag{7.11}$$

By the definition (7.9) of the divergence, for all $\mathcal{Y} \in \mathbb{S}_N$, we have

$$\|\operatorname{div}(\mathcal{Y})\|_F^2 = \sum_{i_1,\ldots,i_N} \left(\sum_{n=1}^{N} (\partial^{(n)}(\mathcal{Y}))_{i_1,\ldots,i_n,\ldots,i_N}\right)^2. \tag{7.12}$$

According to the equation (7.11), we obtain

$$\|\operatorname{div}(\mathcal{Y})\|_F^2 \leqslant 2^{N-1} \sum_{i_1,\ldots,i_N} \sum_{n=1}^{N} \left(\partial^{(n)}(\mathcal{Y})\right)^2. \tag{7.13}$$

On account of the definition (7.8) and by splitting the expression of $\partial^{(n)}(\mathcal{Y})$, we get

$$\begin{aligned}\|\operatorname{div}(\mathcal{Y})\|_F^2 \leqslant 2^{N-1} \sum_{i_1,\ldots,i_N} \Bigg[&\sum_{\substack{n=1 \\ 1<i_n<I_n}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n,\ldots,i_N} - (\mathcal{Y}^{(n)})_{i_1,\ldots,i_n-1,\ldots,i_N}\right)^2 \\ &+ \sum_{\substack{n=1 \\ i_n=1}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n,\ldots,i_N}\right)^2 + \sum_{\substack{n=1 \\ i_n=I_n}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n-1,\ldots,i_N}\right)^2 \Bigg].\end{aligned}$$

Now, using the inequality $(a_1 - a_2)^2 \leqslant 2(a_1^2 + a_2^2)$, we have

$$\begin{aligned}\|\operatorname{div}(\mathcal{Y})\|_F^2 \leqslant 2^{N-1} \sum_{i_1,\ldots,i_N} \Bigg[&2 \sum_{\substack{n=1 \\ 1<i_n<I_n}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n,\ldots,i_N}\right)^2 + 2 \sum_{\substack{n=1 \\ 1<i_n<I_n}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n-1,\ldots,i_N}\right)^2 \\ &+ \sum_{\substack{n=1 \\ i_n=1}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n,\ldots,i_N}\right)^2 + \sum_{\substack{n=1 \\ i_n=I_n}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n-1,\ldots,i_N}\right)^2 \Bigg].\end{aligned}$$

Thus, by regrouping the adequate terms, we get

$$\|\operatorname{div}(\mathcal{Y})\|_F^2 \leqslant 2^{N-1} \sum_{i_1,\ldots,i_N} \left[2 \sum_{\substack{n=1 \\ 1\leqslant i_n<I_n}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n,\ldots,i_N}\right)^2 + 2 \sum_{\substack{n=1 \\ 1<i_n\leqslant I_n}}^{N} \left((\mathcal{Y}^{(n)})_{i_1,\ldots,i_n-1,\ldots,i_N}\right)^2 \right].$$

Now, it is obvious to see that

$$\|\operatorname{div}(\mathcal{Y})\|_F^2 \leqslant 2^{N-1}\left(2\|\mathcal{Y}\|_F^2 + 2\|\mathcal{Y}\|_F^2\right), \tag{7.14}$$

which leads to the main result. □

Remark 7.2.1. *The particular case of this result was proved by Chambolle in [28] in the matrix case which $N = 2$. Namely, we get*

$$\|\operatorname{div}(Y)\|_F^2 \leqslant 8\|Y\|_F^2,\ \forall Y \in \mathbb{R}^{I_1 \times I_2}. \tag{7.15}$$

All the products defined in the tensor space [23, 96, 97] can be very useful to generalize some discrete operations as the discrete gradient and the divergence operation of high-order tensors. Using the n-mode product and its properties (see, [35, 97, 103]), we proved that the partial derivatives $(\nabla_{(n)}\mathcal{X})_n$ of $\mathcal{X}$ can be expressed as follows, for all $n = 1, \ldots, N$,

$$\nabla_{(n)}\mathcal{X} = \mathcal{X} \times_n C_n, \tag{7.16}$$

where $C_n = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & 0 \end{pmatrix} \in \mathbb{R}^{I_n \times I_n}$. The transpose $\nabla_{(n)}^T$ of $\nabla_{(n)}$ is given by

$$\nabla_{(n)}^T\mathcal{X} = \mathcal{X} \times_n C_n^T.$$

According to the relation (7.9), the divergence operator of $\mathcal{Y}$ can be expressed using the n-mode product as follows

$$\operatorname{div}(\mathcal{Y}) = -\sum_{n=1}^{N} \mathcal{Y}^{(n)} \times_n C_n^T. \tag{7.17}$$

The discrete version of the closed convex subset $\mathcal{K}$ is given by

$$\mathcal{K} = \left\{ \mathcal{W} \in \mathbb{S}_N : \sqrt{\sum_{k=1}^{N} |(\mathcal{W}^{(k)})_{i_1,\ldots,i_N}|^2} \leq 1,\ \forall i_1, \ldots, i_N \right\}. \tag{7.18}$$

The convex set $\mathcal{K}$ has $\beta_{\mathcal{K}} = \sqrt{I_1 \cdot I_2 \cdots I_N}$ as the upper bound. While, the discretization of the set Ω will be specified later depending on the chosen model.

Based on Chan, Golub and Mulet [30] and the properties cited in [56], the discrete tensorial total variation of $\mathcal{X}$ can be generalized as follows:

$$TV(\mathcal{X}) = \max_{\mathcal{W} \in \mathcal{K}} \langle \nabla\mathcal{X} \mid \mathcal{W} \rangle, \tag{7.19}$$

which is also the Legendre-Fenchel conjugate of the characteristic function $i_{\mathcal{K}}$ of the

convex set $\mathcal{K}$ [28, 30]. By considering this definition, another formulation of the total variation regularization problem (7.2) is given by

$$\min_{\mathcal{X}\in\Omega}\max_{\mathcal{W}\in\mathcal{K}}\left(\frac{\mu}{2}\|\mathcal{X}-\mathcal{G}\|_F^2+\langle\nabla\mathcal{X}\mid\mathcal{W}\rangle\right). \tag{7.20}$$

Proposition 7.2.1. *The minimization problem (7.20) is equivalent to the following constrained minimax problem*

$$\begin{cases}\max\limits_{\mathcal{W}\in\mathcal{K}}\left(\min\limits_{\substack{\mathcal{X}\in\Omega\\ \mathcal{H}}}\left(\left\|\mathcal{X}-(\mathcal{G}+\frac{1}{\mu}\mathcal{H})\right\|_F^2+\|\mathcal{G}\|_F^2-\left\|\mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W})\right\|_F^2\right)\right).\\ \text{subject to } \mathcal{H}=\operatorname{div}(\mathcal{W}).\end{cases} \tag{7.21}$$

Proof. The function $(\mathcal{X},\mathcal{W})\mapsto\frac{\mu}{2}\|\mathcal{X}-\mathcal{G}\|_F^2-\langle\mathcal{X}\mid\operatorname{div}(\mathcal{W})\rangle$ is convex in $\mathcal{X}$ and concave in $\mathcal{W}$, in addition $\mathcal{K}$ is bounded. Then, according to the *min-max* theorem [131], the above equation can be seen as the dual problem of the *ROF* model (7.2) introduced for the first time in image restoration by Chan, Golub and Mulet [30], and is given by

$$\max_{\mathcal{W}\in\mathcal{K}}\min_{\mathcal{X}\in\Omega}\left(\frac{\mu}{2}\|\mathcal{X}-\mathcal{G}\|_F^2-\langle\mathcal{X}\mid\operatorname{div}(\mathcal{W})\rangle\right). \tag{7.22}$$

On the other hand, the cost function of this optimization problem can be expressed as

$$\frac{\mu}{2}\|\mathcal{X}-\mathcal{G}\|_F^2-\langle\mathcal{X}\mid\operatorname{div}(\mathcal{W})\rangle=\frac{\mu}{2}\left(\left\|\mathcal{X}-(\mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W}))\right\|_F^2+\|\mathcal{G}\|_F^2-\left\|\mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W})\right\|_F^2\right).$$

As a consequence, the problem (7.22) can be written as

$$\max_{\mathcal{W}\in\mathcal{K}}\left(\min_{\mathcal{X}\in\Omega}\left(\left\|\mathcal{X}-(\mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W}))\right\|_F^2+\|\mathcal{G}\|_F^2-\left\|\mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W})\right\|_F^2\right)\right). \tag{7.23}$$

By considering the constraint $\mathcal{H}=\operatorname{div}(\mathcal{W})$ in our problem, it is immediate to see that the optimization problem (7.23) is equivalent to the constrained problem (7.21).

□

7.2.2 Alternating tensorial conditional gradient method

We propose a new approach based on the definition of a pseudo Lagrangian operator associated to the minimization problem (7.21) and the alternating technique that serves in splitting our main problem into feasible subproblems. We consider the functional energy $(\mathcal{X},\mathcal{H},\mathcal{W},\mathcal{R},\mathcal{Q})\longmapsto\mathcal{L}(\mathcal{X},\mathcal{H},\mathcal{W},\mathcal{R},\mathcal{Q})$ as a pseudo Lagrangian functional associated

to the constrained optimization problem (7.21) defined as follows

$$\mathcal{L}(\mathcal{X},\mathcal{H},\mathcal{W},\mathcal{R},\mathcal{Q}) = \left\|\mathcal{X}-(\mathcal{G}+\frac{1}{\mu}\mathcal{H})\right\|_F^2 + \|\mathcal{G}\|_F^2 - \left\|\mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W})\right\|_F^2 + \langle\mathcal{R}\mid\mathcal{H}\rangle - \langle\mathcal{Q}\mid\operatorname{div}(\mathcal{W})\rangle. \tag{7.24}$$

We are going to use an alternating approach that consists in minimizing $\mathcal{L}$ over $\mathcal{X}$ and $\mathcal{H}$, then maximizing $\mathcal{L}$ over $\mathcal{W}$, with respect the linear constraint $\mathcal{H} = \operatorname{div}(\mathcal{W})$. It follows that our optimization problem can be solved approximately by two successive processes: Starting from an initial guess $(\mathcal{X}_0,\mathcal{H}_0,\mathcal{W}_0,\mathcal{R}_0,\mathcal{Q}_0)$, we have first to solve, for all $k \geqslant 0$, the minimization process

$$\begin{cases} 1. \quad \mathcal{H}_{k+1} = \underset{\mathcal{H}}{\operatorname{argmin}}\,\mathcal{L}(\mathcal{X}_k,\mathcal{H},\mathcal{W}_k,\mathcal{R}_k,\mathcal{Q}_k), \\ 2. \quad \mathcal{X}_{k+1} = \underset{\mathcal{X}\in\Omega}{\operatorname{argmin}}\,\mathcal{L}(\mathcal{X},\mathcal{H}_{k+1},\mathcal{W}_k,\mathcal{R}_k,\mathcal{Q}_k), \\ 3. \quad \mathcal{Q}_{k+1} = \dfrac{2}{\mu}\left(\mathcal{X}_{k+1}-\mathcal{G}-\dfrac{1}{\mu}\mathcal{H}_{k+1}\right). \end{cases} \tag{7.25}$$

Using the results of the first process (7.25), then, we solve the second maximization process

$$\begin{cases} 4. \quad \mathcal{W}_{k+1} = \underset{\mathcal{W}\in\mathcal{K}}{\operatorname{argmax}}\,\mathcal{L}(\mathcal{X}_{k+1},\mathcal{H}_{k+1},\mathcal{W},\mathcal{R}_k,\mathcal{Q}_{k+1}). \\ 5. \quad \mathcal{R}_{k+1} = \dfrac{2}{\mu}\left(\mathcal{X}_{k+1}-\mathcal{G}-\dfrac{1}{\mu}\operatorname{div}(\mathcal{W}_{k+1})\right). \end{cases} \tag{7.26}$$

The first problem in (7.25) is equivalent to the following minimization problem

$$\mathcal{H}_{k+1} = \underset{\mathcal{H}}{\operatorname{argmin}}\left(\left\|\mathcal{X}_k-(\mathcal{G}+\frac{1}{\mu}\mathcal{H})\right\|_F^2 + \langle\mathcal{R}_k\mid\mathcal{H}\rangle\right), \tag{7.27}$$

which is clearly given by some algebraic manipulation as

$$\mathcal{H}_{k+1} = \underset{\mathcal{H}}{\operatorname{argmin}}\left\|\mathcal{X}_k-(\mathcal{G}+\frac{1}{\mu}\mathcal{H})-\frac{\mu}{2}\mathcal{R}_k\right\|_F^2. \tag{7.28}$$

Then the iterate $\mathcal{H}_{k+1}$, at each step k, is obtained as

$$\mathcal{H}_{k+1} = \mu\left(\mathcal{X}_k-\mathcal{G}-\frac{\mu}{2}\mathcal{R}_k\right). \tag{7.29}$$

The second problem in (7.25) can be reformulated as

$$\mathcal{X}_{k+1} = \underset{\mathcal{X}\in\Omega}{\operatorname{argmin}}\left\|\mathcal{X}-(\mathcal{G}+\frac{1}{\mu}\mathcal{H}_{k+1})\right\|_F^2, \tag{7.30}$$

which is nothing but the projection of $\mathcal{G}+\frac{1}{\mu}\mathcal{H}_{k+1}$ in the convex set Ω. Thus, the iterate $\mathcal{X}_{k+1}$, at each step k, is given by

$$\mathcal{X}_{k+1} = P_{\Omega}(\mathcal{G}+\frac{1}{\mu}\mathcal{H}_{k+1}), \tag{7.31}$$

where P_{Ω} stands for the projection operator on the convex set Ω. Finally, the maximization problem in (7.26) can be reformulated as the following minimization problem

$$\mathcal{W}_{k+1} = \underset{\mathcal{W}\in\mathcal{K}}{\operatorname{argmin}} \left\| \mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W})+\frac{\mu}{2}\mathcal{Q}_{k+1}\right\|_F^2. \tag{7.32}$$

The constraint minimization subproblem (7.32) is associated to a linear differentiable cost function, which make the resolution more easier. One of the most efficient method that can handle this problem is the conditional gradient method that we will discuss in detail in Algorithm 7.2.2. Finally, we choose the sequences $(\mathcal{R}_k)_k$ and $(\mathcal{Q}_k)_k$ such that the constraint $\mathcal{H}=\operatorname{div}(\mathcal{W})$ is satisfied in every step k, which lead to the expressions of $\mathcal{R}_k$ and $\mathcal{Q}_k$ given in (7.25) and (7.26). Thus, the problems (7.25) and (7.26) can be summarized as follows. Having an initial guess $(\mathcal{X}_0,\mathcal{R}_0)$, we have

$$\forall k\in\mathbb{N}^*,\begin{cases} \mathcal{H}_{k+1} = \mu\left(\mathcal{X}_k-\mathcal{G}-\frac{\mu}{2}\mathcal{R}_k\right),\\ \mathcal{X}_{k+1} = P_{\Omega}\left(\mathcal{G}+\frac{1}{\mu}\mathcal{H}_{k+1}\right),\\ \mathcal{Q}_{k+1} = \frac{2}{\mu}\left(\mathcal{X}_{k+1}-\mathcal{G}-\frac{1}{\mu}\mathcal{H}_{k+1}\right),\\ \mathcal{W}_{k+1} = \underset{\mathcal{W}\in\mathcal{K}}{\operatorname{argmin}}\left\|\mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W})+\frac{\mu}{2}\mathcal{Q}_{k+1}\right\|_F^2,\\ \mathcal{R}_{k+1} = \frac{2}{\mu}\left(\mathcal{X}_{k+1}-\mathcal{G}-\frac{1}{\mu}\operatorname{div}(\mathcal{W}_{k+1})\right). \end{cases} \tag{7.33}$$

7.2.3 Tensorial conditional gradient method

In this section, we deal with the problem (7.32). To solve the constrained minimization $\mathcal{W}$-subproblem in every step k, we are going to use the conditional gradient method [9, 11]. The conditional gradient method, also known as the Frank-Wolfe method [49], is an iterative gradient descent method subject to the convex constraint conditions. This approach depends on an appropriate choice of both the step size and the direction of the descent. First, let us consider, for a fixed k, the following function $F_{\mu,k}$ given by

$$F_{\mu,k}(\mathcal{W}) = \left\| \mathcal{G}+\frac{1}{\mu}\operatorname{div}(\mathcal{W})+\frac{\mu}{2}\mathcal{Q}_{k+1}\right\|_F^2. \tag{7.34}$$

The minimization $\mathcal{W}$-subproblem is nothing but the minimization of $F_{\mu,k}$ over the convex set $\mathcal{K}$, namely,

$$\min_{\mathcal{W}\in\mathcal{K}} F_{\mu,k}(\mathcal{W}). \tag{7.35}$$

Solving the minimization problem (7.35) by a descent gradient approach requires to generate a convergent sequence $(\mathcal{W}_l(k))_l$ to the solution $\mathcal{W}^*(k)$ of (7.35). The definition of this sequence based on two essential elements: The direction and the step size of the descent. Thus, the expression of $(\mathcal{W}_l(k))_l$ at each step l is given by

$$\mathcal{W}_{l+1}(k) = \mathcal{W}_l(k) + \alpha_l \mathcal{D}_l,\ \forall l \geqslant 0. \tag{7.36}$$

The tensor sequence $(\mathcal{D}_l)_l$ represents the direction of the descent that must be orthogonal to the residue of the solution, and the positive parameter α_l denotes the step size of the descent. For the sake of notation simplicity, we denote in the rest of this section $\mathcal{W}_l(k)$ by $\mathcal{W}_l$.

The choice of the direction of the descent as a minimization direction stems of the first order approximation of the cost function $F_{\mu,k}$. The objective function $F_{\mu,k}$ is Gateau differentiable and its gradient is given by

$$\nabla F_{\mu,k}(\mathcal{W}) = -\frac{2}{\mu}\nabla\left(\mathcal{G} + \frac{1}{\mu}\operatorname{div}(\mathcal{W}) + \frac{\mu}{2}\mathcal{Q}_{k+1}\right).$$

Then, for a given $\mathcal{W}_l$ at the step l, the first order in the Taylor formula is given by

$$F_{\mu,k}(\mathcal{W}) = F_{\mu,k}(\mathcal{W}_l) + \left\langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \mathcal{W} - \mathcal{W}_l \right\rangle + o(\|\mathcal{W} - \mathcal{W}_l\|_F).$$

According to Lemma 7.2.1, we have

$$o(\|\mathcal{H}\|_F) = \|\frac{1}{\mu}\operatorname{div}(\mathcal{H})\|_F^2 \leq \frac{2^{N+1}}{\mu^2}\|\mathcal{H}\|_F^2. \tag{7.37}$$

Thus, the minimization problem (7.35) can be approached, at each step l, by the following linear constrained optimization problem

$$\min_{\mathcal{W}\in\mathcal{K}} \left\langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \mathcal{W} - \mathcal{W}_l \right\rangle. \tag{7.38}$$

Let $\widehat{\mathcal{W}_l}$ be the solution of (7.38). It is then clear that the line connecting $\widehat{\mathcal{W}_l}$ and $\mathcal{W}_l$ lies entirely inside the convex set $\mathcal{K}$, which make a feasible direction of the descent $\mathcal{D}_l$ to be considered, as

$$\mathcal{D}_l = \widehat{\mathcal{W}_l} - \mathcal{W}_l.$$

In accordance with the overall result of the projection calculation provided in [11], it is

immediate to see that the solution of (7.38) is given by

$$\widehat{\mathcal{W}_l} = P_{\mathcal{K}}\left(\frac{-\beta_{\mathcal{K}}\nabla F_{\mu,k}(\mathcal{W}_l)}{\|\nabla F_{\mu,k}(\mathcal{W}_l)\|_F}\right), \tag{7.39}$$

where $P_{\mathcal{K}}$ is the projection on the closed convex set $\mathcal{K}$ and $\beta_{\mathcal{K}} = \sqrt{I_1 \cdots I_N}$ is an upper bound of $\mathcal{K}$. Let us notice that in our context, the projection $P_{\mathcal{K}}$ on the convex set $\mathcal{K}$ is given for all column block tensor $\mathcal{Y} = (\mathcal{Y}^{(1)}, \ldots, \mathcal{Y}^{(N)}) \in \mathbb{S}_N$ by

$$P_{\mathcal{K}}(\mathcal{Y}) = \mathcal{Z} := (\mathcal{Z}^{(1)}, \ldots, \mathcal{Z}^{(N)}) \text{ where } (\mathcal{Z}^{(n)})_{i_1,i_2,\ldots,i_N} = \frac{(\mathcal{Y}^{(n)})_{i_1,i_2,\ldots,i_N}}{\max\{1, |\mathcal{Y}_{i_1,i_2,\ldots,i_N}|\}}, \ \forall n. \tag{7.40}$$

By definition of the gradient algorithm, the step size sequence $(\alpha_l)_l$ is defined, at each step l, as the minimizer of $\xi(\alpha) = F_{\mu,k}(\mathcal{W}_l + \alpha\mathcal{D}_l)$, namely,

$$\alpha_l^* = \underset{0\leqslant\alpha\leqslant 1}{\operatorname{argmin}} F_{\mu,k}(\mathcal{W}_l + \alpha\mathcal{D}_l). \tag{7.41}$$

It is a quadratic unidimensional problem in which the solution is given by

$$\alpha_l^* = \begin{cases} \alpha_l & if \quad 0 \leqslant \alpha_l \leqslant 1, \\ 1 & if \quad \alpha_l > 1, \end{cases} \tag{7.42}$$

where the positive scalar α_l is given by $\alpha_l = -\dfrac{\mu^2}{2}\dfrac{\left\langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \mathcal{D}_l \right\rangle}{\|\operatorname{div}(\mathcal{D}_l)\|_F}$.

The tensorial version of the conditional gradient method to solve the minimization problem (7.35), at each step k, can be summarized in Algorithm 7.1.

Algorithm 7.1 Tensorial conditional gradient algorithm.

1: **Input:** $\mathcal{G}$, $\mathcal{Q}_{k+1}$, μ, $\beta_{\mathcal{K}}$, ϵ.
2: **Initialization:** choose $\mathcal{W}_0 = \mathcal{W}_k$.
3: **while** $\|\mathcal{W}_{l+1} - \mathcal{W}_l\|_F / \|\mathcal{W}_l\|_F \geqslant \epsilon$ **do**
4: **Compute:** $\mathcal{V}_l = \nabla F_{\mu,k}(\mathcal{W}_l)$.
5: **Compute:** $\widehat{\mathcal{W}_l} = P_{\mathcal{K}}\left(-\beta_{\mathcal{K}}\dfrac{\mathcal{V}_l}{\|\mathcal{V}_l\|_F}\right)$ using the definition (7.40).
6: **Update:** The direction of descent $\mathcal{D}_l = \widehat{\mathcal{W}_l} - \mathcal{W}_l$.
7: **Compute:** The step size α_l^* verifies (7.42).
8: **Compute:** $\mathcal{W}_{l+1} = \mathcal{W}_l + \alpha_l^*\mathcal{D}_l$.
9: **end while**
10: **return** $\mathcal{W}_{k+1}$

To boost the accuracy of our algorithm and decrease the cost of the computation, two remarks can be very useful. First, we notice that the proposed approach to solve our

problem (7.21) is an iterative algorithm that contains another iterative algorithm which is the conditional gradient algorithm 7.1. This process can be very expensive in the term of the execution time. Second, we observe that the structure of our algorithm enables a restarted version to be used. Recall that the restarted process consists in repeating the algorithm by initializing the input $\mathcal{X}_0$ with the resulting estimation computed in the previous one [136]. This technique can be very efficient in some applications to increase the precision of our algorithm. Therefore, we propose to combine the two iterative methods into one which is described as an alternating tensorial conditional gradient algorithm (*ATCG-TV*) given in Algorithm 7.2.

Algorithm 7.2 Alternating tensorial conditional gradient algorithm (ATCG-TV)

Input: $\mathcal{G}, \mathcal{X}_0 = \mathcal{G}, \mathcal{R}_0 = 0, \mu, \epsilon$.
while $\|\mathcal{W}_{k+1} - \mathcal{W}_k\|_F / \|\mathcal{W}_k\|_F \geqslant \epsilon$ **do**
  Compute: $\mathcal{H}_{k+1} = \mu\left(\mathcal{X}_k - \mathcal{G} - \frac{\mu}{2}\mathcal{R}_k\right)$.
  Compute: $\mathcal{X}_{k+1} = P_\Omega\left(\mathcal{G} + \frac{1}{\mu}\mathcal{H}_{k+1}\right)$.
  Compute: $\mathcal{Q}_{k+1} = \frac{2}{\mu}\left(\mathcal{X}_{k+1} - \mathcal{G} - \frac{1}{\mu}\mathcal{H}_{k+1}\right)$.
  Compute: $\widehat{\mathcal{W}_k} = P_{\mathcal{K}}\left(-\beta_{\mathcal{K}} \frac{\nabla F_{\mu,k}(\mathcal{W}_k)}{\|\nabla F_{\mu,k}(\mathcal{W}_k)\|}\right)$ using the definition (7.40).
  Update: The direction of descent $\mathcal{D}_k = \widehat{\mathcal{W}_k} - \mathcal{W}_k$.
  Compute: The step size α_k^* computed in (7.42).
  Compute: $\mathcal{W}_{k+1} = \mathcal{W}_k + \alpha_k^* \mathcal{D}_k$.
  Update: $\mathcal{R}_{k+1} = \frac{2}{\mu}\left(\mathcal{X}_{k+1} - \mathcal{G} - \frac{1}{\mu}\operatorname{div}(\mathcal{W}_{k+1})\right)$.
end while
return $\mathcal{X}_{k+1}$.

7.2.4 Convergence results

In this section, we establish some results on the convergence of the sequence generated by the proposed algorithms.

Theorem 7.2.1. *At a fixed step k, the sequence $(\mathcal{W}_l)_l$ generated by Algorithm 7.1 is a minimizing sequence, i.e.,*

$$\lim_{l\to\infty} F_{\mu,k}(\mathcal{W}_l) = \min_{\mathcal{W}\in\mathcal{K}} F_{\mu,k}(\mathcal{W}) = F_{\mu,k}(\mathcal{W}^*(k)).$$

Proof. From the convexity of the function $F_{\mu,k}$, we have

$$F_{\mu,k}(\mathcal{W}^*(k)) - F_{\mu,k}(\mathcal{W}) \geqslant \left\langle \nabla F_{\mu,k}(\mathcal{W}) \mid \mathcal{W}^*(k) - \mathcal{W} \right\rangle, \ \forall \mathcal{W} \in \mathcal{K}.$$

where $\mathcal{W}^*(k) \in \mathcal{K}$ is a solution of the problem (7.35). It follows that

$$0 \leqslant F_{\mu,k}(\mathcal{W}_l) - F_{\mu,k}(\mathcal{W}^*(k)) \leqslant -\left\langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \mathcal{W}^*(k) - \mathcal{W}_l \right\rangle. \tag{7.43}$$

Since $\widehat{\mathcal{W}}_l$ is a solution of the problem (7.38), then, we have

$$0 \leqslant F_{\mu,k}(\mathcal{W}_l) - F_{\mu,k}(\mathcal{W}^*(k)) \leqslant -\left\langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \widehat{\mathcal{W}}_l - \mathcal{W}_l \right\rangle. \tag{7.44}$$

Therefore, $\langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \widehat{\mathcal{W}}_l - \mathcal{W}_l \rangle \leq 0$. As a consequence, we have

$$0 \leqslant F_{\mu,k}(\mathcal{W}_l) - F_{\mu,k}(\mathcal{W}^*(k)) \leqslant \left| \langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \widehat{\mathcal{W}}_l - \mathcal{W}_l \rangle \right|. \tag{7.45}$$

Thus, the proof is completed by showing that $\lim_{l \to \infty} \langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \widehat{\mathcal{W}}_l - \mathcal{W}_l \rangle = 0$.

Let us set $\eta_l = \langle \nabla F_{\mu,k}(\mathcal{W}_l) \mid \widehat{\mathcal{W}}_l - \mathcal{W}_l \rangle$ and $\gamma_l(\alpha) = \mathcal{W}_l + \alpha\left(\widehat{\mathcal{W}}_l - \mathcal{W}_l\right)$, for any $\alpha \in [0,1]$. The Taylor expansion of $F_{\mu,k}$ at $\mathcal{W}_l$ is given by

$$F_{\mu,k}(\gamma_l(\alpha)) - F_{\mu,k}(\mathcal{W}_l) = \alpha\eta_l + o\left(\alpha \left\| \widehat{\mathcal{W}}_l - \mathcal{W}_l \right\|_F\right).$$

According to (7.37), we have

$$\left| o\left(\alpha \left\| \hat{\mathcal{W}}_l - \mathcal{W}_l \right\|_F\right) \right| \leqslant \frac{2^{N+1}\alpha^2}{\mu^2} \left\| \widehat{\mathcal{W}}_l - \mathcal{W}_l \right\|_F^2.$$

Now, using the triangular inequality together with the inequality $(a_1 + a_2)^2 \leqslant 2(a_1^2 + a_2^2)$, we may write the following inequalities

$$\begin{aligned} \|\hat{\mathcal{W}}_l - \mathcal{W}_l\|_F^2 &\leqslant \left(\|\hat{\mathcal{W}}_l\|_F + \|\mathcal{W}_l\|_F\right)^2 \\ &\leqslant 2\left(\|\hat{\mathcal{W}}_l\|_F^2 + \|\mathcal{W}_l\|_F^2\right) \\ &\leqslant 4\beta_{\mathcal{K}}^2. \end{aligned} \tag{7.46}$$

We recall that $\beta_{\mathcal{K}}$ denotes the upper bound of the convex set $\mathcal{K}$. Then, for all $\alpha \in [0,\ 1]$ and for all positive integers l,

$$F_{\mu,k}(\gamma_l(\alpha)) - F_{\mu,k}(\mathcal{W}_l) \leqslant \alpha\eta_l + \frac{2^{N+3}\alpha^2\beta_{\mathcal{K}}^2}{\mu^2}. \tag{7.47}$$

From Algorithm 7.1, we have

$$\mathcal{W}_{l+1} = \mathcal{W}_l + \alpha_l^*\left(\widehat{\mathcal{W}}_l - \mathcal{W}_l\right) = \gamma_l\left(\alpha_l^*\right),$$

where α_l^* is a solution of the problem (7.41). It is clear that

$$\forall \alpha \in [0,\ 1],\ \forall l, \quad F_{\mu,k}(\mathcal{W}_{l+1}) \leqslant F_{\mu,k}(\gamma_l(\alpha)). \tag{7.48}$$

In particular, for $\alpha = 0$, we obtain

$$F_{\mu,k}(\mathcal{W}_{l+1}) \leqslant F_{\mu,k}(\gamma_l(0)) = F_{\mu,k}(\mathcal{W}_l).$$

In addition, we have

$$F_{\mu,k}(\mathcal{W}_l) \geqslant \min_{\mathcal{W}\in\mathcal{K}} F_{\mu,k}(\mathcal{W}).$$

It follows that the sequence $\left(F_{\mu,k}(\mathcal{W}_l)\right)_l$ is monotonically decreasing and bounded, which means that the sequence $\left(F_{\mu,k}(\mathcal{W}_l)\right)_l$ is convergent. Consequently,

$$\lim_{l\to\infty}\left(F_{\mu,k}(\mathcal{W}_l) - F_{\mu,k}(\mathcal{W}_{l+1})\right) = 0. \tag{7.49}$$

From (7.47)-(7.48) and the fact that $\eta_l \leqslant 0$, we have, for all $\alpha \in [0,1]$ and for all positive integers l,

$$F_{\mu,k}(\mathcal{W}_l) - F_{\mu,k}(\mathcal{W}_{l+1}) \geqslant \alpha\left|\eta_l\right| - \frac{2^{N+3}\alpha^2\beta_{\mathcal{K}}^2}{\mu^2}.$$

For all $\alpha \in (0,1]$ and for all positive integers l this gives

$$0 < \left|\eta_l\right| \leqslant \frac{2^{N+3}\alpha^2\beta_{\mathcal{K}}^2}{\mu^2} + \frac{F_{\mu,k}(\mathcal{W}_l) - F_{\mu,k}(\mathcal{W}_{l+1})}{\alpha}.$$

According to (7.49), letting $l \to \infty$, we obtain

$$0 \leqslant \liminf_{l\to\infty}\left|\eta_l\right| \leqslant \limsup_{l\to\infty}\left|\eta_l\right| \leqslant \frac{2^{N+3}\alpha^2\beta_{\mathcal{K}}^2}{\mu^2}, \quad \forall \alpha \in (0,1].$$

Finally, we take, in the last inequality, $\alpha \to 0$, we obtain the desired result.

□

Theorem 7.2.2. *We assume that the sequence $(\mathcal{W}_k)_k$ converge at least linearly to a limit $\mathcal{W}^* \in \mathcal{K}$. Then, the sequence $(\mathcal{H}_k, \mathcal{X}_k, \mathcal{W}_k, \mathcal{R}_k, \mathcal{Q}_k)_k$ generated by Algorithm 7.2 is convergent.*

Proof. We assume that the sequence $(\mathcal{W}_k)_k$ converge linearly to a limit $\mathcal{W}^*$, then there exist a scalar $0 < r < 1$, such that

$$\|\mathcal{W}_{k+1} - \mathcal{W}_k\|_F \leqslant r\|\mathcal{W}_k - \mathcal{W}_{k-1}\|_F,\ \forall k \geqslant k_0. \tag{7.50}$$

where $k_0 > 0$ is some positive integer. On account of to Algorithm 7.2 and Lemma 7.2.1,

we have

$$\begin{aligned}\|\mathcal{H}_{k+1}-\mathcal{H}_k\|_F &= \|\operatorname{div}(\mathcal{W}_{k+1})-\operatorname{div}(\mathcal{W}_k)\|_F,\\ &\leqslant 2^{N+1}\|\mathcal{W}_{k+1}-\mathcal{W}_k\|_F.\end{aligned}$$

Then, it follows that

$$\|\mathcal{H}_{k+1}-\mathcal{H}_k\|_F \leqslant r^k 2^{N+1}\|\mathcal{W}_1-\mathcal{W}_0\|_F. \tag{7.51}$$

As a consequence, for two integer $n \geqslant m$, we have

$$\begin{aligned}\|\mathcal{H}_n-\mathcal{H}_m\|_F &\leqslant \sum_{k=m}^{n-1}\|\mathcal{H}_{k+1}-\mathcal{H}_k\|_F,\\ &\leqslant 2^{N+1}\|\mathcal{W}_1-\mathcal{W}_0\|_F\sum_{k=m}^{n-1} r^k,\end{aligned}$$

Then, we have

$$\|\mathcal{H}_n-\mathcal{H}_m\|_F \leqslant 2^{N+1}\|\mathcal{W}_1-\mathcal{W}_0\|_F\frac{r^m-r^n}{1-r}, \tag{7.52}$$

by making m tend to $+\infty$, we deduce that $(\mathcal{H}_k)_k$ is a Cauchy sequence on $\mathbb{T}_N$. Thus, by a consequence, the sequence $(\mathcal{H}_k)_k$ converge to a limit $\mathcal{H}^* \in \mathbb{T}_N$.

Likewise and according to the equation defining the iterate $\mathcal{X}_k$ in Algorithm 7.2, it is immediate to prove the convergence of the sequence $(\mathcal{X}_k)_k$ to a limit $\mathcal{X}^*$. Thereafter, it is clear that the sequences $(\mathcal{R}_k)_k$ and $(\mathcal{Q}_k)_k$ converge to the same limit $\frac{2}{\mu}\left(\mathcal{X}^*-\mathcal{G}-\frac{1}{\mu}\operatorname{div}(\mathcal{W}^*)\right)$.

□

7.3 An accelerated tensorial double proximal gradient method

The main goal of this section is the resolution of the constrained tensorial total variation minimization problem (7.1). Our contribution is threefold. Firstly, a gradient descent-like algorithm is developed to minimize the non-differentiable and non-linear total variation problem over a convex set by computing first the proximal mapping of the total variation term and projecting after the problem using Tseng's Splitting Algorithm [4]. Secondly, since the gradient algorithms have a slow convergence rate, we will accelerate our proposed algorithm using some extrapolation techniques. Finally, such methods are represented in the tensorial representation which may extend the range of application of our model and developed algorithm.

Let first consider the following closed proper convex functions

$$\begin{aligned} \mathscr{F} : \mathbb{R}^{I_1\times\cdots\times I_N} &\longrightarrow \mathbb{R}_+ \\ \mathcal{X} &\longrightarrow \mathscr{F}(\mathcal{X}) = \frac{1}{2}\|\mathscr{H}(\mathcal{X}) - \mathcal{B}\|_F^2, \\ \mathscr{G} : \mathbb{R}^{I_1\times\cdots\times I_N} &\longrightarrow \mathbb{R}_+ \\ \mathcal{X} &\longrightarrow \mathscr{G}(\mathcal{X}) = \mu|||\nabla\mathcal{X}|||_1. \end{aligned}$$

The gradient operator of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ is given in Proposition 6.2.1. As in the classical vectorial case, the function $\mathscr{F}$ is differentiable, and its gradient on $\mathcal{X}$ is given by

$$\nabla\mathscr{F}(\mathcal{X}) = \mathscr{H}^*(\mathscr{H}(\mathcal{X}) - \mathcal{B}), \tag{7.53}$$

and the function $\mathscr{G}$ is not differentiable due to the non differentiability of the l^1 norm, which make the resolution of this minimization problem more complex.

Let $f : \mathbb{R}^{I_1\times\cdots\times I_N} \longrightarrow \mathbb{R}\cup\{\infty\}$ be a closed proper convex function [127]. We recall the proximal operator of f in the following definition.

Definition 7.3.1. *[127] The **proximal operator** (also called the proximal mapping) of f is the operator given by*

$$prox_f(\mathcal{U}) = \underset{\mathcal{X}}{\operatorname{argmin}}\left(f(\mathcal{X}) + \frac{1}{2}\|\mathcal{X}-\mathcal{U}\|^2\right) \text{ for any } \mathcal{U} \text{ in } \mathbb{R}^{I_1\times\cdots\times I_N}. \tag{7.54}$$

Since the cost function of the minimization problem defined above is strongly convex and not everywhere infinite, then, there exist a unique minimizer for every $\mathcal{U}$ in $\mathbb{R}^{I_1\times\cdots\times I_N}$, see [4, 127] for more details. We will often encounter the proximal operator of the scaled function λf with $\lambda > 0$, which can be expressed as

$$prox_{\lambda f}(\mathcal{U}) = \underset{\mathcal{X}}{\operatorname{argmin}}\left(f(\mathcal{X}) + \frac{1}{2\lambda}\|\mathcal{X}-\mathcal{U}\|^2\right). \tag{7.55}$$

The operator $prox_{\lambda f}$ is also called the proximal operator of f with the parameter λ.

Definition 7.3.2. *The convex conjugate of f is given by*

$$f^*(\mathcal{Y}) = \sup_{\mathcal{X}}\left(\langle\mathcal{X}|\mathcal{Y}\rangle - f(\mathcal{X})\right), \ \forall\mathcal{Y} \in \mathbb{R}^{I_1\times\cdots\times I_N}. \tag{7.56}$$

The convex constrained minimization problem (7.1) can be written as:

$$\min_{\mathcal{X}\in\Omega}\left(\mathscr{F}(\mathcal{X}) + \mathscr{G}(\mathcal{X})\right), \tag{7.57}$$

where Ω is a convex nonempty bounded set. As $\mathscr{F}$ and $\mathscr{G}$ are proper lower semicontinuous convex functions, $\mathscr{F}$ is Gateau differentiable and uniformly convex on $\mathbb{R}^{I_1\times\cdots\times I_N}$ and if it is further assumed that Ω is closed, then, there exists a unique solution of the minimization problem (7.57), see [4, 132, 38, 67, 43] for a deeper discussion.

7.3.1 Accelerated tensorial double proximal gradient algorithm

Now, we will introduce an interesting extension of gradient descent method to handle this tensorial convex minimization problem. In the literature, the gradient descent technique is developed in a variety of ways to handle different minimization problems, such as nonlinear minimization problems [38], fractional optimization problems [18] and others. The proximal gradient method represents a generalized form of the gradient descent method in the presence of non differentiability in the cost function [3, 4, 127, 132]. First, we consider the unconstrained minimization problem

$$\min_{\mathcal{X}\in\mathbb{R}^{I_1\times\cdots\times I_N}} (\mathscr{F}(\mathcal{X})+\mathscr{G}(\mathcal{X})). \tag{7.58}$$

Suppose that, at the step k, we have constructed an iterate tensor $\mathcal{X}_k$ that approximate the solution of the constrained minimization problem (7.57). The quadratic approximation of $\mathscr{F}$ based at the iterate tensor $\mathcal{X}_k$, for $\lambda_k > 0$, is given by

$$\Phi_k(\mathcal{X}) = \mathscr{F}(\mathcal{X}_k) + \langle \mathcal{X}-\mathcal{X}_k | \nabla\mathscr{F}(\mathcal{X}_k)\rangle + \frac{1}{2\lambda_k}\|\mathcal{X}-\mathcal{X}_k\|_F^2. \tag{7.59}$$

Then, it is immediate to see that the problem (7.58) is approached, at each step k, by the following minimization problem

$$\min_{\mathcal{X}}\left(\mathscr{G}(\mathcal{X}) + \frac{1}{2\lambda_k}\|\mathcal{X}-\mathcal{X}_k+\lambda_k\nabla\mathscr{F}(\mathcal{X}_k)\|_F^2\right). \tag{7.60}$$

According to Definition 7.3.1, the minimization problem (7.60) admits a unique minimizer $\mathcal{Z}_k$ given by

$$\mathcal{Z}_k = prox_{\lambda_k\mathscr{G}}(\mathcal{X}_k - \lambda_k\nabla\mathscr{F}(\mathcal{X}_k)),\ \forall k\in\mathbb{N}, \tag{7.61}$$

where the operator $prox_{\lambda_k\mathscr{G}}$ denotes the proximal mapping of $\mathscr{G}$ with the parameter λ_k.

In general, the algorithm proposed for computing $\mathcal{Z}_k$ by (7.61) required two essential elements. The first one is an optimal selection of the step size sequence $(\lambda_k)_k$ that depend on the Lipschitz constant of $\nabla\mathscr{F}$ (to be discussed later), and the second one is the computation of the proximal operator of $\lambda_k\mathscr{G}$ which is given in the following proposition.

Proposition 7.3.1. *For all $\mathcal{Y}\in\mathbb{R}^{I_1\times\cdots\times I_N}$ and $\lambda>0$, the proximal operator(mapping) of $\lambda\mathscr{G}$ is given by*

$$\mathcal{Z} = prox_{\lambda\mathscr{G}}(\mathcal{Y}) = \mathcal{Y} + \lambda\nabla^T(\mathcal{P}^*), \tag{7.62}$$

where $\mathcal{P}^*$ is an optimal solution of

$$\min_{\mathcal{P}}\left(\mathscr{G}_1^*(-\mathcal{P}) + \langle\frac{\lambda}{2}\nabla(\nabla^T\mathcal{P}) + \nabla\mathcal{Y}|\mathcal{P}\rangle\right), \tag{7.63}$$

with the function $\mathscr{G}_1^*$ is the conjugate function of $\mathscr{G}_1 := \mu|||\,.\,|||_1$.

Proof. By Definition 7.3.1, for all $\mathcal{Y}$ and $\lambda>0$, $prox_{\lambda\mathscr{G}}(\mathcal{Y})$ is the the unique optimal

solution of following unconstrained minimization problem

$$\min_{\mathcal{U}}\left(\mathcal{G}(\mathcal{U})+\frac{1}{2\lambda}\|\mathcal{U}-\mathcal{Y}\|_F^2\right). \tag{7.64}$$

If we assume that $\mathcal{G}(\mathcal{U})=\mathcal{G}_1(\nabla\mathcal{U})$, with $\mathcal{G}_1(\mathcal{Y})=\mu|||\mathcal{Y}|||_1$, the minimization problem can be transformed to a constrained minimization problem as follow

$$\begin{cases}\min\limits_{\mathcal{U},\mathcal{V}}\left(\mathcal{G}_1(\mathcal{V})+\frac{1}{2\lambda}\|\mathcal{U}-\mathcal{Y}\|_F^2\right)\\ s.t \qquad \mathcal{V}=\nabla\mathcal{U}.\end{cases} \tag{7.65}$$

The Lagrangian function associated with this problem is defined as

$$\mathcal{L}(\mathcal{U},\mathcal{V},\mathcal{P})=\mathcal{G}_1(\mathcal{V})+\frac{1}{2\lambda}\|\mathcal{U}-\mathcal{Y}\|_F^2+\langle\mathcal{P}|\mathcal{V}-\nabla\mathcal{U}\rangle. \tag{7.66}$$

As a consequence, the solution of (7.65) is exactly the saddle point of $\mathcal{L}$ (see [57]), which is the solution of the Lagrangian primal problem

$$\min_{\mathcal{U},\mathcal{V}}\max_{\mathcal{P}}\mathcal{L}(\mathcal{U},\mathcal{V},\mathcal{P}). \tag{7.67}$$

Since the Lagrangian is separable with respect to $\mathcal{U}$ and $\mathcal{V}$, then we may switch the min-max order based on the min-max theorem [43, 132]. As a consequence, the Lagrangian dual problem can be written as

$$\max_{\mathcal{P}}\left[\min_{\mathcal{U}}\left(\frac{1}{2\lambda}\|\mathcal{U}-\mathcal{Y}\|_F^2-\langle\mathcal{P}|\nabla\mathcal{U}\rangle\right)+\min_{\mathcal{V}}\left(\mathcal{G}_1(\mathcal{V})+\langle\mathcal{P}|\mathcal{V}\rangle\right)\right]. \tag{7.68}$$

On the one hand, it is clear that the minimizer of the problem in $\mathcal{U}$ is given by $\mathcal{U}^*=\mathcal{Y}+\lambda\nabla^T\mathcal{P}$ with a corresponding optimal value equal to:

$$\begin{aligned}\min_{\mathcal{U}}\left(\frac{1}{2\lambda}\|\mathcal{U}-\mathcal{Y}\|_F^2-\langle\mathcal{P}|\nabla\mathcal{U}\rangle\right)&=\frac{1}{2\lambda}\|\mathcal{U}^*-\mathcal{Y}\|_F^2-\langle\mathcal{P}|\nabla\mathcal{U}^*\rangle\\ &=-\langle\frac{\lambda}{2}\nabla(\nabla^T\mathcal{P})+\nabla\mathcal{Y}|\mathcal{P}\rangle.\end{aligned} \tag{7.69}$$

On the other hand, the second minimization problem verifies

$$\min_{\mathcal{V}}\left(\mathcal{G}_1(\mathcal{V})+\langle\mathcal{P}|\mathcal{V}\rangle\right)=-\max_{\mathcal{V}}\left(\langle-\mathcal{P}|\mathcal{V}\rangle-\mathcal{G}_1(\mathcal{V})\right)=-\mathcal{G}_1^*(-\mathcal{P}), \tag{7.70}$$

where we recall that $\mathcal{G}_1^*$ is the convex conjugate function of $\mathcal{G}_1$. As a result, we obtain the following dual problem

$$\max_{\mathcal{P}}\left[-\mathcal{G}_1^*(-\mathcal{P})-\langle\frac{\lambda}{2}\nabla(\nabla^T\mathcal{P})+\nabla\mathcal{Y}|\mathcal{P}\rangle\right], \tag{7.71}$$

that can be rewritten as the minimization problem (7.63).

□

So far, we have shown that $prox_{\lambda\mathscr{G}}(\mathcal{Y}) = \mathcal{Y} + \lambda\nabla^T\mathcal{P}^*$, where $\mathcal{P}^*$ is an optimal solution of the minimization problem (7.63). In other words, the calculation of the proximal mapping of the function $\mathscr{G}$ required, at each iteration k, the resolution of the minimization problem (7.63). For $\mathcal{Y} = \mathcal{Y}_k := \mathcal{X}_k - \lambda_k\nabla\mathscr{F}(\mathcal{X}_k)$, and $\lambda = \lambda_k$, at the step k, we have to solve the problem

$$\min_{\mathcal{P}}\Big(\mathscr{G}_1^*(-\mathcal{P}) + \langle\frac{\lambda_k}{2}\nabla(\nabla^T\mathcal{P}) + \nabla\mathcal{Y}_k|\mathcal{P}\rangle\Big). \tag{7.72}$$

For all $k \in \mathbb{N}$, let us consider the operators $\mathscr{K}$ and $(\mathscr{D}_k)_k$ defined as

$$\begin{array}{rcl}\mathscr{K} : \mathbb{S}_N & \longrightarrow & \mathbb{R}_+\\ \mathcal{P} & \longrightarrow & \mathscr{K}(\mathcal{P}) = \mathscr{G}_1^*(-\mathcal{P}),\\ \mathscr{D}_k : \mathbb{S}_N & \longrightarrow & \mathbb{R}\\ \mathcal{P} & \longrightarrow & \mathscr{D}_k(\mathcal{P}) = \langle\frac{\lambda_k}{2}\nabla(\nabla^T\mathcal{P}) + \nabla\mathcal{Y}_k|\mathcal{P}\rangle.\end{array}$$

Since the objective functional of the minimization problem (7.72) is a sum of closed proper convex function $\mathscr{K}$, and closed proper convex differentiable function $\mathscr{D}_k$, then, we can use again the proximal gradient approach to solve (7.72). Hence, the solution can be approximated by the sequence $(\mathcal{P}_l)_l$ defined as

$$\forall l \in \mathbb{N}, \quad \mathcal{P}_{l+1} = prox_{\alpha_l\mathscr{K}}(\mathcal{P}_l - \alpha_l\nabla\mathscr{D}_k(\mathcal{P}_l)), \tag{7.73}$$

with $\alpha_l > 0$ is a step size parameter. Notice that the expression of the proximal gradient (7.73) required two important ingredients, the gradient of the differentiable function $\mathscr{D}_k$ and the proximal mapping of the non differentiable function $\mathscr{K}$. It is immediate to see that the gradient of $\mathscr{D}_k$ is given by

$$\forall k, \quad \nabla\mathscr{D}_k(\mathcal{P}) = \lambda_k\nabla(\nabla^T\mathcal{P}) + \nabla\mathcal{Y}_k. \tag{7.74}$$

On the other hand, the proximal mapping of $prox_{\alpha_l\mathscr{K}}$ is discussed in the following proposition.

Proposition 7.3.2. *For all $\mathcal{P} \in \mathbb{S}_N$, the proximal mapping of $\alpha_l\mathscr{K}$ is given by*

$$prox_{\alpha_l\mathscr{K}}(\mathcal{P}) = \mathcal{P} + prox_{\alpha_l\mathscr{G}_1}(-\mathcal{P}),\ \forall l \in \mathbb{N}, \tag{7.75}$$

where $\mathscr{G}_1 := \mu|||\,.\,|||_1$.

Proof. For any $\mathcal{P} \in \mathbb{S}_N$, we have

$$\begin{aligned} prox_{\alpha_l \mathscr{K}}(\mathcal{P}) &= \underset{\mathcal{W}}{\operatorname{argmin}} \left(\mathscr{K}(\mathcal{W}) + \frac{1}{2\alpha_l} \|\mathcal{W} - \mathcal{P}\|_F^2 \right), \\ &= \underset{\mathcal{W}}{\operatorname{argmin}} \left(\mathscr{G}_1^*(-\mathcal{W}) + \frac{1}{2\alpha_l} \|\mathcal{W} - \mathcal{P}\|_F^2 \right), \\ &= -\underset{\mathcal{V}}{\operatorname{argmin}} \left(\mathscr{G}_1^*(\mathcal{V}) + \frac{1}{2\alpha_l} \|\mathcal{V} + \mathcal{P}\|_F^2 \right), \\ &= -prox_{\alpha_l \mathscr{G}_1^*}(-\mathcal{P}). \end{aligned}$$

According to Moreau decomposition [118], we have the following property that relate the *prox* operator of any proper closed convex function f by their conjugates

$$prox_f(x) + prox_{f^*}(x) = x, \quad \forall x. \tag{7.76}$$

Using the relation (7.76), we obtain the desired conclusion

$$prox_{\alpha_l \mathscr{K}}(\mathcal{P}) = \mathcal{P} + prox_{\alpha_l \mathscr{G}_1}(-\mathcal{P}). \tag{7.77}$$

Furthermore, the tensorial proximal mapping $prox_{\alpha_l \mathscr{G}_1} := prox_{\alpha_l \mu \|\|\|\cdot\|\|\|_1}$ of the function $\mathscr{G}_1$ is a direct result of the proximal operator of the $\|\|\|\cdot\|\|\|_1$ norm, also known as the soft thresholding operator in the vector case [127]. Then, by using the Moreau's formula (7.76) and the fact that the $\|\|\|\cdot\|\|\|_\infty$ norm is the dual norm of the $\|\|\|\cdot\|\|\|_1$ norm, thus, the proximal operator $prox_{\eta \|\|\|\cdot\|\|\|_1}$, with any $\eta > 0$, can be computed based on the orthogonal projection on the $\|\|\|\cdot\|\|\|_\infty$-unit ball [127], which is the unit box. This leads to

$$\left(prox_{\eta \|\|\|\cdot\|\|\|_1}(\mathcal{P})\right)_{i_1,\ldots,i_N} = \begin{cases} \mathcal{P}_{i_1,\ldots,i_N} - \eta, & \mathcal{P}_{i_1,\ldots,i_N} \geqslant \eta, \\ 0, & |\mathcal{P}_{i_1,\ldots,i_N}| < \eta, \\ \mathcal{P}_{i_1,\ldots,i_N} + \eta, & \mathcal{P}_{i_1,\ldots,i_N} \leqslant -\eta, \end{cases} \tag{7.78}$$

which establishes the formula and ends the proof. □

Finally, the algorithm (7.61) computing the sequence $(\mathcal{X}_k)_k$ can be summarized as the following double iterative algorithm

$$\forall k \in \mathbb{N}, \begin{cases} \forall l = 1, \cdots, l_k, \begin{cases} \mathcal{Q}_l = \mathcal{P}_l - \alpha_l \nabla \mathscr{D}_k(\mathcal{P}_l), \\ \mathcal{P}_{l+1} = \mathcal{Q}_l + prox_{\alpha_l \mathscr{G}_1}(-\mathcal{Q}_l), \end{cases} \\ \mathcal{Y}_k = \mathcal{X}_k - \lambda_k \nabla \mathscr{F}(\mathcal{X}_k), \\ \mathcal{Z}_k = \mathcal{Y}_k + \lambda_k \nabla^T(\mathcal{P}_{l_k+1}), \end{cases} \tag{7.79}$$

where the step size scalar sequences $(\lambda_k)_k$ and $(\alpha_l)_l$ depend on the Lipschitz constants $L(\nabla \mathscr{F})$ and $L(\nabla \mathscr{D}_k)$, respectively, if they are exist.

In the following subsection, we will compute the approximated solution $\mathcal{X}_{k+1}$ of the constrained minimization problem (7.57) at the step $k+1$ by projecting the iterate $\mathcal{Z}_k$ in the convex set Ω using the Tseng's splitting approach.

7.3.2 Tseng's Splitting Algorithm

Tseng's splitting algorithm proposed in [4] considers as a modified version of the proximal gradient algorithm (7.61) used to handle the constrained convex nonsmooth optimization problem. Under a nonempty closed and convex constraint Ω, the algorithm may given as follows:

$$\forall k \in \mathbb{N}, \quad \begin{cases} \mathcal{Y}_k = \mathcal{X}_k - \lambda_k \nabla \mathscr{F}(\mathcal{X}_k), \\ \mathcal{Z}_k = prox_{\lambda_k \mathscr{G}}(\mathcal{Y}_k), \\ \mathcal{R}_k = \mathcal{Z}_k - \lambda_k \nabla \mathscr{F}(\mathcal{Z}_k), \\ \mathcal{X}_{k+1} = \Pi_\Omega(\mathcal{X}_k - \mathcal{Y}_k + \mathcal{R}_k). \end{cases} \tag{7.80}$$

where Π_Ω denotes the orthogonal projection on the convex set Ω.

Theorem 7.3.1. *Let $\mathcal{X}^*$ denotes the unique solution of the problem (7.57). Suppose that Ω is furthermore a closed set in $\mathbb{R}^{I_1 \times \cdots \times I_N}$. Then, the sequence $(\mathcal{X}_k)_k$ generated by Algorithm 7.80 satisfies $\|\mathcal{X}_k - \mathcal{X}^*\|_F \longrightarrow 0$ as $k \longrightarrow \infty$.*

Proof. As the functional $\mathscr{F}$ and $\mathscr{G}$ are proper lower semicontinuous convex functions, $\mathscr{F}$ is Gateau differentiable and uniformly convex on $\mathbb{R}^{I_1 \times \cdots \times I_N}$ and the set Ω is a closed convex nonempty subset of $\mathbb{R}^{I_1 \times \cdots \times I_N}$, then, the strong convergence of the sequence $(\mathcal{X}_k)_k$ is an immediate consequence of the general result in [4] (see Pr.27.13 pg. 407). □

7.3.3 The step size parameters selection

The choice of the step size parameters is considered as a typical condition that ensures the convergence of the sequence $(\mathcal{X}_k)_k$ to the minimizer of the problem (7.57). It is required that the values of the step size parameters λ_k and α_l be in the intervals $(0, \frac{1}{L(\nabla \mathscr{F})})$ and $(0, \frac{1}{L(\nabla \mathscr{D}_k)})$, respectively, where $L(\nabla \mathscr{F})$ and $L(\nabla \mathscr{D}_k)$ denote the Lipschitz constants of the operators $\nabla \mathscr{F}$ and $\nabla \mathscr{D}_k$, respectively (see [4, 38] for further details).

For all couple $(\mathcal{X}, \mathcal{Y})$ in $\mathbb{R}^{I_1 \times \cdots \times I_N} \times \mathbb{R}^{I_1 \times \cdots \times I_N}$, we have

$$\begin{aligned} \|\nabla \mathscr{F}(\mathcal{X}) - \nabla \mathscr{F}(\mathcal{Y})\|_F &= 2\|\mathscr{H}^T(\mathscr{H}(\mathcal{X})) - \mathscr{H}^T(\mathscr{H}(\mathcal{Y}))\|_F, \\ &= 2\|\mathscr{H}^T(\mathscr{H}(\mathcal{X} - \mathcal{Y}))\|_F, \\ &\leqslant 2|||\mathscr{H}^T \circ \mathscr{H}|||.\|\mathcal{X} - \mathcal{Y}\|_F, \end{aligned}$$

where $\circ$ stands for the composition operation. Then, we may choose as a Lipschitz constant of the operator $\nabla \mathscr{F}$ the constant

$$L(\nabla \mathscr{F}) = 2|||\mathscr{H}^T \circ \mathscr{H}|||. \tag{7.81}$$

As consequence, the step size $(\lambda_k)_k$ can be chosen as a fixed step size value $\lambda_k := \lambda \in (0, \frac{1}{L(\nabla \mathscr{F})})$.

In the case of the operator $\nabla \mathscr{D}_k$, for a fixed step k, a Lipschitz constant $L(\nabla \mathscr{D}_k)$ is not known, then, the step sizes (α_l) can be found by a line search method [127], which mean

that we apply the proximal gradient method with an easy backtracking stepsize rule as fallows

$$\forall l \in \mathbb{N}, \quad \alpha_l = \tau \alpha_{l-1}, \tag{7.82}$$

where the scalar $\tau > 0$ is a line search parameter.

7.3.4 Accelerated tensorial double proximal gradient algorithm

It is well known that the proximal gradient algorithm suffers from a slow convergence rate. We will present in this section an accelerated version of the proximal gradient algorithm consists in adding an extrapolation step in the algorithm, in order to compute the solution in less steps than the basic proximal gradient. A large amount of research has been conducted to different extrapolation algorithms applied to a variety of general problems [24, 85, 138, 32, 112], and others developed of the proximal gradient method [5, 122].

Definition 7.3.3. *Let $(\mathcal{X}_k)_k$ and $(\mathcal{T}_k)_k$ two convergent sequences to the same limit $\mathcal{X}^*$, we say that $(\mathcal{T}_k)$ converges faster than the sequence $(\mathcal{X}_k)$ if*

$$\lim_{k \longrightarrow \infty} \frac{\|\mathcal{T}_k - \mathcal{X}_*\|}{\|\mathcal{X}_k - \mathcal{X}_*\|} = 0. \tag{7.83}$$

The goal of the extrapolation is to find a sequence $(\mathcal{T}_k)_k$ from the sequence $(\mathcal{X}_k)_k$ so that $(\mathcal{T}_k)_k$ converges faster to the same limit as $(\mathcal{X}_k)_k$. There are many extrapolation methods in the literature, but we will only be interested to apply the Nesterov's algorithm approach and the polynomial extrapolation methods to our tensorial nonlinear minimization problem.

Tensorial Nesterov acceleration techniques

One simple and widely studied strategy is to perform extrapolation in the spirit of Nesterov's extrapolation techniques [107, 122]. The basic idea of this technique is to make use of historical information at each iteration in order to reduce the convergence rate from $O(1/k)$ to $O(1/k^2)$. Thus, using the position of the current iteration tensor and the previous iteration tensor, the tensorial double proximal gradient method is accelerated by adding an extrapolation step given by

$$\mathcal{T}_{k+1} = \mathcal{X}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathcal{X}_k - \mathcal{X}_{k-1}), \tag{7.84}$$

where the scalars (t_k) is given, at each step k, by $t_{k+1} = \dfrac{\sqrt{4t_k^2 + 1} + 1}{2}$.

The convergence of the sequence $(\mathcal{T}_k)_{k \in \mathbb{N}}$ may be investigated by following the approaches given in several papers, e.g [38, 37]. The complete algorithm that summarizes

Accelerated tensorial double proximal gradient (ATDPG) with Nesterov's extrapolation is given in Algorithm 7.3.

Algorithm 7.3 *ATDPG*-Nesterov algorithm

Inputs: Initial guess $\mathcal{X}_1 = \mathcal{O}$, $\mathcal{P}_1 = \mathcal{O}$, $\mathcal{T}_1 = \mathcal{X}_0$, $\nabla\mathscr{F}$, $t_1 = 1$, τ, μ, λ, α_0, tol.
for $k = 1, \ldots$ until convegence, **do**
  $\mathcal{Y}_k = \mathcal{T}_k - \lambda\nabla\mathscr{F}(\mathcal{T}_k)$,
  Compute the operator $\nabla\mathscr{D}_k$ using the formula (7.74),
  for $l = 1, \ldots, l_k$ **do**
    Update the line search parameter α_l using (7.82).
    $\mathcal{Q}_l = \mathcal{P}_l - \alpha_l\nabla\mathscr{D}_k(\mathcal{P}_l)$,
    $\mathcal{P}_{l+1} = \mathcal{Q}_l + prox_{\alpha_l\mu\|\|\|.\|\|\|_1}(-\mathcal{Q}_l)$, with $prox_{\alpha_l\mu\|\|\|.\|\|\|_1}$ is given in (7.78).
  end for
  $\mathcal{Z}_k = \mathcal{Y}_k + \lambda\nabla^T(\mathcal{P}_{l_k+1})$,
  $\mathcal{R}_k = \mathcal{Z}_k - \lambda\nabla\mathscr{F}(\mathcal{Z}_k)$,
  $\mathcal{X}_k = \Pi_\Omega(\mathcal{T}_k - \mathcal{Y}_k + \mathcal{R}_k)$.
  $t_{k+1} = \dfrac{\sqrt{4t_k^2 + 1} + 1}{2}$,
  Compute $\mathcal{T}_{k+1} = \mathcal{X}_k + \left(\dfrac{t_k - 1}{t_{k+1}}\right)(\mathcal{X}_k - \mathcal{X}_{k-1})$.
  End the iteration if $\|\mathcal{T}_{k+1} - \mathcal{T}_k\|_F / \|\mathcal{T}_k\|_F < tol$.
end for
return $\mathcal{T}_{k+1}$.

The global tensorial polynomial extrapolation methods

The polynomial extrapolation methods are among the best known extrapolation methods thanks to their theoretical clarity and numerical efficiency, especially when applied to solving nonlinear problems such as the case of our main problem (7.1). The polynomial extrapolation methods was introduced in [138] for the vectorial extrapolation case that developed after in [17, 87, 85] using efficient implementation techniques. Those methods was also developed in a matrix global form in [84], and recently was generalized for the tensor sequences in [44] using tensor product. In the spirit of [85], we define the transformation in the following form

$$\mathcal{T}_{k,q} = \sum_{j=0}^{q} \gamma_j \mathcal{X}_{k+j}, \tag{7.85}$$

where k defines the first term of the sequence, the integer q stands for the number of terms of the sequence, and the scalars (γ_j) verify the following two conditions

$$\begin{cases} \sum_{j=0}^{q} \gamma_j = 1\,, \\ \sum_{j=0}^{q} \gamma_j \langle \mathcal{V}_i | \Delta\mathcal{X}_{k+j} \rangle = 0, \quad i = 0, 1, \ldots, q-1, \end{cases} \tag{7.86}$$

where $\Delta\mathcal{X}_j = \mathcal{X}_{j+1} - \mathcal{X}_j$ and $\mathcal{V}_i$ are given tensors. As a consequence, the conditions (7.86) lead to the following linear system

$$\begin{cases} \gamma_0 & + & \gamma_1 & + & \cdots & + & \gamma_q & = 1 \\ \gamma_0 \langle \mathcal{V}_0 | \Delta\mathcal{X}_k \rangle & + & \gamma_1 \langle \mathcal{V}_0 | \Delta\mathcal{X}_{k+1} \rangle & + & \cdots & + & \gamma_q \langle \mathcal{V}_0 | \Delta\mathcal{X}_{k+q} \rangle & = 0 \\ \gamma_0 \langle \mathcal{V}_1 | \Delta\mathcal{X}_k \rangle & + & \gamma_1 \langle \mathcal{V}_1 | \Delta\mathcal{X}_{k+1} \rangle & + & \cdots & + & \gamma_q \langle \mathcal{V}_1 | \Delta\mathcal{X}_{k+q} \rangle & = 0 \\ \vdots & & \vdots & & & & \vdots & \vdots \\ \gamma_0 \langle \mathcal{V}_{q-1} | \Delta\mathcal{X}_k \rangle & + & \gamma_1 \langle \mathcal{V}_{q-1}, \Delta\mathcal{X}_{k+1} \rangle & + & \cdots & + & \gamma_q \langle \mathcal{V}_{q-1} | \Delta\mathcal{X}_{k+q} \rangle & = 0 \end{cases} \tag{7.87}$$

where the vector $\boldsymbol{\gamma} = [\gamma_0, \gamma_1, \ldots, \gamma_q]^T$ is the solution of the matrix equation

$$\underbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \langle \mathcal{V}_0 | \Delta\mathcal{X}_k \rangle & \langle \mathcal{V}_0 | \Delta\mathcal{X}_{k+1} \rangle & \cdots & \langle \mathcal{V}_0 | \Delta\mathcal{X}_{k+q} \rangle \\ \langle \mathcal{V}_1 | \Delta\mathcal{X}_k \rangle & \langle \mathcal{V}_1 | \Delta\mathcal{X}_{k+1} \rangle & \cdots & \langle \mathcal{V}_1 | \Delta\mathcal{X}_{k+q} \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle \mathcal{V}_{q-1} | \Delta\mathcal{X}_k \rangle & \langle \mathcal{V}_{q-1} | \Delta\mathcal{X}_{k+1} \rangle & \cdots & \langle \mathcal{V}_{q-1} | \Delta\mathcal{X}_{k+q} \rangle \end{pmatrix}}_{M} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_q \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{7.88}$$

It is clear that $\mathcal{T}_{k,q}$ exists and is unique if and only if the square matrix M is nonsingular, which depends on the choice of $(\mathcal{V}_i)$ tensors.

Remark 7.3.1. *In the polynomial vector extrapolation case, the author in [138, 85] proposed a numerically stable and computationally economical algorithm for computing the $(\gamma_i)_i$ via the QR factorization. The same concept was developed for the tensor case in [44] by defining a new QR factorization of the tensor $\mathcal{U}_k$ contents the sequence $(\Delta\mathcal{X}_i)$ as frontal slices. However, in our situation, we have only used a direct method for solving the matrix equation (7.88).*

In this work, we consider two polynomial extrapolation methods, Global tensor minimal polynomial extrapolation (GT-MPE), where the sequence $\mathcal{V}_i$ is defined as

$$\mathcal{V}_i = \Delta\mathcal{X}_{i+k}, \tag{7.89}$$

and Global tensor Reduced Rank Extrapolation (GT-RRE) with

$$\mathcal{V}_i = \Delta^2\mathcal{X}_{i+k} = \Delta\mathcal{X}_{i+k+1} - \Delta\mathcal{X}_{i+k}. \tag{7.90}$$

Both methods generate a sequence $\mathcal{T}_{k,q}$ which approximates the solution under the condition that q is smaller than an integer q_0. Where q_0 is an integer such that

$\left\{\Delta\mathcal{X}_0, \Delta\mathcal{X}_1, \ldots, \Delta\mathcal{X}_{q_0-1}\right\}$ is a linearly independent set, but $\left\{\Delta\mathcal{X}_0, \Delta\mathcal{X}_1, \ldots, \Delta\mathcal{X}_{q_0-1}, \Delta\mathcal{X}_{q_0}\right\}$ is not. For more details, we refer the reader to [138, 85]. The process of polynomial extrapolation using *GT-MPE* or *GT-RRE* is summarized in Algorithm 7.4.

Algorithm 7.4 The *GT-MPE*/*GT-RRE* algorithm

1: **Inputs:** The sequence $\left\{\mathcal{X}_k, \cdots, \mathcal{X}_{k+q+1}\right\}$.
2: Compute M:
3: For *GT-MPE*: $\mathcal{V}_i = \Delta\mathcal{X}_{i+k}$.
4: For *GT-RRE*: $\mathcal{V}_i = \Delta^2\mathcal{X}_{i+k}$.
5: Solve the matrix equation $M\boldsymbol{\gamma} = e_1$.
6: Compute the approximation $\mathcal{T}_{k,q}$ using the expression (7.85).
7: **Output:** $\mathcal{T}_{k,q}$.

Note that for the *GT-MPE* and *GT-RRE* methods, the number of calculations required increases quadratically with the number of iterations q and the storage cost increases linearly. A good method to keep the cost of storage and the cost of the lowest possible calculations is to restart these algorithms periodically. The following algorithm describes the restarted version of those methods called also a cycling mode.

Algorithm 7.5 Cycling mode.

1: Fix the integer q.
2: Form the sequence $\left\{\mathcal{X}_1, \cdots, \mathcal{X}_{q+1}\right\}$.
3: Calculate the approximation $\mathcal{T}_{k,q}$ using the algorithm 7.4.
4: If $\mathcal{T}_{k,q}$ is satisfactory, stop. Otherwise, set $\mathcal{X}_k = \mathcal{T}_{k,q}$ as a new initialization and $k = k+1$, and go to the second step.

The accelerated version of the tensorial double proximal gradient algorithm using the polynomial method is summarized in Algorithm 7.6.

7.4 Conclusion

In the presence of a convex constraint, projected gradient methods are considered the most widely used for solving differentiable problems. In this chapter, we have developed two extensions of these techniques to deal with the non-differentiable total variation regularization problem in a tensorial context. The alternating conditional gradient method designed for a particular case of the problem, while the accelerated double proximal gradient algorithm is designed for a more general case.

Algorithm 7.6 *ATDPG*-Polynomial extrapolation algorithm

Initial guess $\mathcal{X}_0 = \mathcal{O}$, $\mathcal{P}_0 = \mathcal{O}$ $\mathcal{T}_1 = \mathcal{X}_0$, $\nabla\mathcal{F}$, μ, λ, τ, α_0, q, tol.
for $k = 1, \ldots$ until convegence, **do**
  $\mathcal{Y}_k = \mathcal{T}_k - \lambda_k \nabla\mathcal{F}(\mathcal{T}_k)$,
  Compute the operator $\nabla\mathcal{D}_k$ using the formula (7.74),
  for $l = 1, \ldots, l_k$ **do**
    Update the line search parameter α_l using (7.82).
    $\mathcal{Q}_l = \mathcal{P}_l - \alpha_l \nabla\mathcal{D}_k(\mathcal{P}_l)$,
    $\mathcal{P}_{l+1} = \mathcal{Q}_l + prox_{\alpha_l \mu \|\|\cdot\|\|_1}(-\mathcal{Q}_l)$, with $prox_{\alpha_l \mu \|\|\cdot\|\|_1}$ is given in (7.78).
  end for
  $\mathcal{Z}_k = \mathcal{Y}_k + \lambda \nabla^T(\mathcal{P}_{l_k+1})$,
  $\mathcal{R}_k = \mathcal{Z}_k - \lambda \nabla\mathcal{F}(\mathcal{Z}_k)$,
  $\mathcal{X}_k = \Pi_\Omega(\mathcal{T}_k - \mathcal{Y}_k + \mathcal{R}_k)$.
  Compute the iterate $\mathcal{T}_{k+1}$ using Algorithm 7.5.
  End the iteration if $\|\mathcal{T}_{k+1} - \mathcal{T}_k\|_F / \|\mathcal{T}_k\|_F < tol$
end for
return $\mathcal{T}_{k+1}$.

CHAPTER 8

Applications to color image and video processing

8.1 Introduction

In the current chapter, our goal is to illustrate the efficiency of all the developed algorithms by applying them to the regularization of the inverse tensor problem related to color images and video processing. From a simple observation based on the definition of the frontal slices of a third order tensor (see Figure 4.1), we can represent a color image by a third-order tensor $\mathcal{X}$ of size $height \times width \times 3$, as illustrate in Figure 8.1.

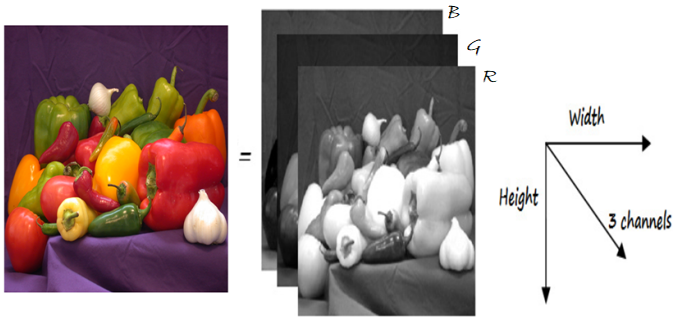


Figure 8.1: "peppers.png"color image in tensor format.

Similarly, the "traffic.avi"color video can be represented by a fourth-order tensor $\mathcal{Y}$ of size $height \times width \times 3 \times frames$.

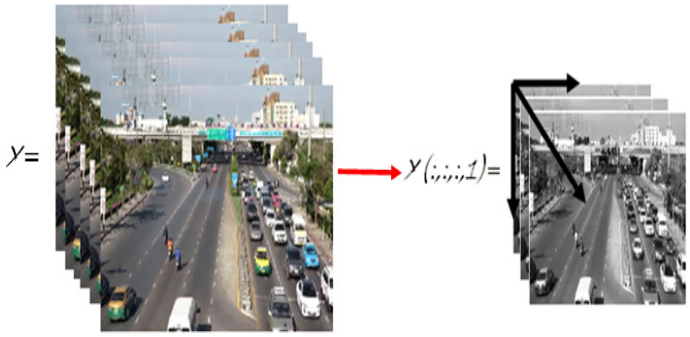


Figure 8.2: "traffic.avi"color video in tensor format.

Image Processing is a technique to enhance digital images from unwanted degradations. Image processing is used in various applications such as remote sensing, medical imaging, the film industry, document processing, printing industry, etc. The common process used in image processing is image compressing, storing, enhancing, and completing. The models that come from the analysis of these problems are generally ill-posed problems. In this chapter, we will apply our regularization algorithms in two subfields which are restoration and completion of color images and videos.

To evaluate the effectiveness of our algorithm, we will use some tools that measure the quality of recovered tensors, such as, Signal to Noise Ratio (SNR), Peak Signal to Noise Ratio (PSNR) and the relative error (RE). By definition, the SNR function gives an objective evaluation between the original image $\mathcal{X}_{true}$ and the resulting image $\mathcal{X}^*$, and it is defined as follows:

$$SNR = 10\log_{10}\left(\frac{\|\mathcal{X}_{true} - E(\mathcal{X}_{true})\|_F^2}{\|\mathcal{X}_{true} - \mathcal{X}^*\|_F^2}\right),$$

where $E(\mathcal{X}_{true})$ denotes the mean of the original image $\mathcal{X}_{true}$. For the same purpose, the $PSNR$ is given by

$$PSNR = 10\log_{10}\left(\frac{d^2 I_1 \cdots I_N}{\|\mathcal{X}_{true} - \mathcal{X}^*\|_F^2}\right), \tag{8.1}$$

where $I_1 \times \cdots \times I_N$ is the size of $\mathcal{X}_{true}$ and d is the maximal variation in the input data. The only difference between the SNR and the $PSNR$ is that the SNR is defined in relation to the signal while the $PSNR$ is defined in relation to the maximum dynamic range. The relative error (RE) is given by

$$RE = \frac{\|\mathcal{X}_{true} - \mathcal{X}^*\|_F}{\|\mathcal{X}_{true}\|_F}.$$

On the other hand, we adopt the stopping criterion of the algorithm by

$$e_k = \frac{\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F}{\|\mathcal{X}_k\|_F} \leqslant tol,$$

where the tolerance tol is chosen in $\{10^{-3}, 10^{-2}\}$.

In image applications, many operations require algebraic manipulation of pixel values, where it may be necessary to allow non-integer values. Then, before performing arithmetic operations, we will convert images and videos to double precision and normalize the intensity to fall in the range of $[0, 1]$. All computations were carried out using the MATLAB R2020a environment on an Intel(R) Core(TM) i5-4670 CPU @ 3.40 GHz computer with 16 GB of RAM.

8.2 Color image and video restoration

Image restoration refers to removal or minimization of known degradations in an image. Figure 8.3 shows a typical situation consisting of two blocks: The degradation process that is modeled as a degradation function H that together with an additive noise term n, operates on an input image f to produce a degraded image g. The second block represents the restoration process : Given degraded image g with some knowledge about the degradation function H, and the additive noise term n, the objective of restoration is to obtain an estimate $\hat{f}$ of the original image.

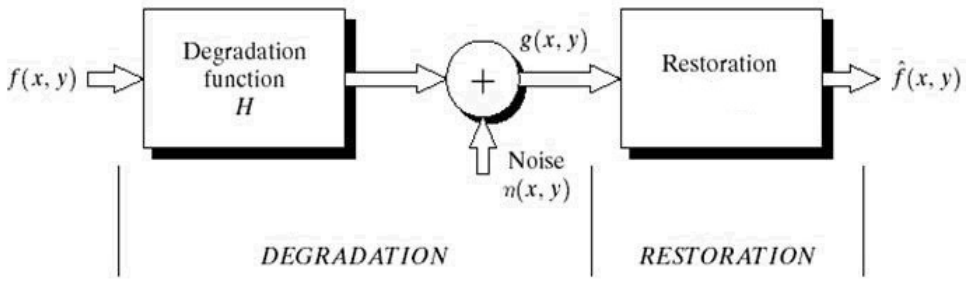


Figure 8.3: Degradation and restoration processes.

In the literature, the function that describes the blurring is called the point spread function (PSF) [74, 21]. The properties of this function give us a clear vision about the degradation model. For example, if the point spread function (PSF) is spatially invariant, which means that the image is blurred in exactly the same way at every spatial location, then, the degradation process is described as a matrix-vector equation:

$$g = Hf + \eta, \tag{8.2}$$

where g is $n \cdot m$ vector representing the observed image of size $m \times n$, f is a vector representing the true image, and η is a vector representing the noise. H is $n \cdot m \times n \cdot m$ blurring square matrix. In addition, if the PSF is separable, then the blurring matrix can

be decomposed as a Kronecker product of two blurring matrices of appropriate sizes[74, 88]. In the case of nonseparable *PSF*, one can use a Kronecker product approximation (*KPA*) to decompose the matrix H as a sum of Kronecker product [82, 144]. Suppose that the matrix H is given as a sum of Kronecker product as follow

$$H = \sum_{i=1}^{r} H_2^{(i)} \otimes H_1^{(i)},$$

where r is the rank of the *PSF* array. As a consequence of the following ***vec*** operator property

$$\boldsymbol{vec}(AXB) = (B^T \otimes A)\boldsymbol{vec}(X), \tag{8.3}$$

our degradation process will be written as

$$\sum_{i=1}^{r} H_1^{(i)} X H_2^{(i)\,T} + N = G, \tag{8.4}$$

where $f = \boldsymbol{vec}(X)$, $g = \boldsymbol{vec}(G)$, and $\eta = \boldsymbol{vec}(N)$. Unfortunately, the form (8.4) is not adapted for color image or video. In order, to obtain a unified and adapted version for color image and video, we have to transform our restoration model into the tensorial format. For that purpose, it is first necessary to develop a new degradation process. The main idea is based on the tensor frontal slices notion and the matrix equation (8.4). Our proposed deblurring tensorial model is given in the following proposition.

Proposition 8.2.1. *Let $\mathcal{X} \in \mathbb{R}^{m_1 \times \cdots \times m_N}$ be a tensor representing a color image or a video.*

1. *Assume that the blurring matrix H is spatially invariant and separable, then the degradation operator will be expressed as follows:*

$$\mathscr{H}(\mathcal{X}) = \mathcal{X} \times_1 H_1 \times_2 H_2. \tag{8.5}$$

2. *In particular, for third order tensor, if we suppose that the blur is **different for each channel/frame** and separable, then the degradation operator will be written as:*

$$\mathscr{H}(\mathcal{X}) = \sum_{n=1}^{N} \mathcal{X} \times_1 H_1^n \times_2 H_2^n \times_3 E_n, \tag{8.6}$$

where $(H_i^n)_{i,n}$ represent the blurring matrices, and $(E_n)_n$ is a sequence of matrices where each E_n denotes an $m_3 \times m_3$ sparse matrix contains zero except the (n,n) position contains 1.

Proof.

1. In the first point, we assume that the blur is spatially invariant which means that the blurring matrix H is the same for all pixels. In addition, we also suppose that

we have the same point spread function (*PSF*). Hence, by using the frontal slices $(\boldsymbol{X}_i)_i$ and $(\boldsymbol{G}_i)_i$ of tensors $\mathcal{X}$ and $\mathcal{G}$ respectively, the blur is modeled by

$$\begin{cases} H\boldsymbol{vec}(\boldsymbol{X}_1) &= \boldsymbol{vec}(\boldsymbol{G}_1) \\ H\boldsymbol{vec}(\boldsymbol{X}_2) &= \boldsymbol{vec}(\boldsymbol{G}_2) \\ \vdots & \vdots \\ H\boldsymbol{vec}(\boldsymbol{X}_M) &= \boldsymbol{vec}(\boldsymbol{G}_M) \end{cases} \tag{8.7}$$

where $M = m_3 \cdots m_N$. Since the blur is separable, the matrix H can be written as $H = H_2 \otimes H_1$ the Kronecker product of the matrices $H_2 \in \mathbb{R}^{m_2 \times m_2}$ and $H_1 \in \mathbb{R}^{m_1 \times m_1}$. Then, the equation (8.7) is equivalent to

$$\begin{cases} H_1 \boldsymbol{X}_1 H_2^T &= \boldsymbol{G}_1 \\ H_1 \boldsymbol{X}_2 H_2^T &= \boldsymbol{G}_2 \\ \vdots & \vdots \\ H_1 \boldsymbol{X}_M H_2^T &= \boldsymbol{G}_M \end{cases}$$

which is exactly

$$H_1 [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_M](I_M \otimes H_2)^T = [\boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_M]. \tag{8.8}$$

As a consequence, the equation (8.8) can be expressed as

$$H_1 \mathcal{X}_{(1)}(I_M \otimes H_2^T) = \mathcal{G}_{(1)},$$

where $\mathcal{X}_{(1)}, \mathcal{G}_{(1)} \in \mathbb{R}^{m_1 \times m_2 \cdot M}$ denote the 1-mode matricization of tensors $\mathcal{X}$ and $\mathcal{G}$, respectively. Using the tensor proprieties [97], the degradation process can be written as the following equation:

$$\mathcal{X} \times_1 H_1 \times_2 H_2 \times_3 I_M = \mathcal{G} \tag{8.9}$$

which is exactly the equation (8.5).

2. In the second case, we suppose that the blurring function is different for each frontal slice. By assuming that the blur in each slice is spatially invariant and separable, our degradation process will be written as:

$$\begin{cases} H_1^1 \boldsymbol{X}_1 (H_2^1)^T &= \boldsymbol{G}_1, \\ H_1^2 \boldsymbol{X}_2 (H_2^2)^T &= \boldsymbol{G}_2, \\ \vdots & \vdots \\ H_1^{m_3} \boldsymbol{X}_{m_3} (H_2^{m_3})^T &= \boldsymbol{G}_{m_3}. \end{cases} \tag{8.10}$$

Let $(E_n)_n$ be a sequence of matrices in $\mathbb{R}^{m_3 \times m_3}$, where each matrix E_n contains zero except the (n, n) position contains 1. Then, the equation (8.10) will be equivalent

to:

$$\sum_{n=1}^{m_3} H_1^n \left[\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{m_3}\right](E_n \otimes (H_2^n)^T) = \left[\boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_{m_3}\right]. \tag{8.11}$$

Passing by the definition of the 1-mode matricization, the equation (8.11) can be expressed as

$$\sum_{n=1}^{m_3} H_1^n \mathcal{X}_{(1)} (E_n \otimes H_2^n)^T = \mathcal{G}_{(1)},$$

which leads to the tensorial degradation process:

$$\sum_{n=1}^{m_3} \mathcal{X} \times_1 H_1^n \times_2 H_2^n \times_3 E_n = \mathcal{G}.$$

□

The developed approaches are based on the prior knowledge of some tools such as blurring matrices and noise error, which are unknown in practice. Thus, having some knowledge about the degradation process is essential for a more optimal restoration. Without knowing at least approximate parameters of the blur, the restoration processes show poor results and then the image will be rather distorted much more than restored.

As already illustrated in Figure 8.3, the procedure is divided into two parts: the degradation where we study the properties of our model, such as the blur approximation, etc. Then, the restoration where we exploit all this information. Starting from no information other than the observed image, blind restoration approaches can be used to approximate the degradation model and restore the image at the same time [31, 47, 20]. Among the most interesting methods in this context, we find the Richardson-Lucy algorithm based on the Fourier domain techniques [47]. The Richardson-Lucy algorithm is known for its implementation of the maximum likelihood and its ability to approximate the *PSF* in the presence of high noise levels. However, numerical tests show that this algorithm provides an optimal approximation of the *PSF* but returns a poor restored image. For that purpose, after an identification of the degradation model proprities, the proposed techniques can be applied.

In general, degradation can be described by two elements: blur and noise. Blur can be defined as a transition in the original image caused by various factors such as movement between the camera and an object, atmospheric turbulence, camera defocusing, etc. Which gives rise to different types of noise (Gaussian, motion, out-of-focus,...)[21]. On the other hand, noise in images can be seen as a variation of brightness or color information. There exist various noise sources, for example, it can be produced at the time of capturing an image through a camera or during transmission.

Color image and video restoration problems provide an application where a tensorial linear equation of the form

$$\mathcal{X} \times_1 H_1 \times_2 H_2 = \tilde{\mathcal{B}} + \mathcal{N}, \tag{8.12}$$

is to be solved, where the coefficient matrices H_1, H_2 represent the blurring operation,

the tensor $\mathcal{X}$ is the color image or video to be reconstructed, and the added error to the right-hand side tensor $\tilde{\mathcal{B}}$, also called additive noise, is represented by tensor $\mathcal{N}$.

For instance, the blurring matrices can be generated by the Toeplitz matrix $H_{\sigma,d}$ that models a uniform blur, with a variance σ, and a band d, and it is given by

$$(H_{\sigma,d})_{ij} = \begin{cases} \dfrac{1}{\sigma\sqrt{2\pi}} \exp\left(-\dfrac{(i-j)^2}{2\sigma^2}\right), & |i-j| \leqslant d, \\ 0 & otherwise. \end{cases}$$

In addition, other blurring operation can be found in [74]. On the other hand, the noise can be in general additive or multiplicative [83]. Gaussian white noise is known to be an additive noise with normally distributed random entries with zero mean and with variance σ that usually is chosen to correspond to a specific noise level $\nu := \|\mathcal{N}\|_F / \|\hat{\mathcal{B}}\|_F$. Salt and pepper noise or impulsive noise appears in the images as dark pixels in bright regions and bright pixels in dark regions [83].

8.2.1 Tensorial Tikhonov regularization algorithm

Tensorial Tikhonov regularization Algorithm 5.2 will be used to restore color images and videos. We set the regularization operator $\mathscr{R}$ using the regularization matrices R_1 and R_2 defined as a bidiagonal matrix $R_1 = diag(-1,1)$, and the matrix R_2 as the identity matrix. Since our tensorial Tikhonov regularization problem (5.3), is a minimization problem under a constraint Ω, then, we worked with two examples of the set Ω, either the box $\Omega_1 = B(\mathcal{L},\mathcal{U})$ with lower tensor bound $\mathcal{L}$ and the upper tensor bound $\mathcal{U}$, or the closed ball $\Omega_2 = \bar{B}_r$ of radius $r = \sqrt{n_1 \cdot n_2 \cdots n_N}$ and of center 0. In order to define local smoothing constraint, we determine the bound tensors $\mathcal{L}$ and $\mathcal{U}$ from the parameters that describe the local properties of our data [76, 19]. For a pixel of the degraded data $\mathcal{B}$, the local mean and variance at $\{i_1,\ldots,i_N\}$ over a $3^N = 3\times 3\times\ldots\times 3$ window are defined as

$$\mathcal{M}_{\mathcal{B}}(i_1,\ldots,i_N) = \frac{1}{3^N} \sum_{k_1=i_1-1}^{i_1+1} \cdots \sum_{k_N=i_N-1}^{i_N+1} \mathcal{B}(k_1,\ldots,k_N),$$

$$\mathcal{V}_{\mathcal{B}}(i_1,\ldots,i_N) = \frac{1}{3^N} \sum_{k_1=i_1-1}^{i_1+1} \cdots \sum_{k_N=i_N-1}^{i_N+1} (\mathcal{B}(k_1,\ldots,k_N) - \mathcal{M}_{\mathcal{B}}(k_1,\ldots,k_N))^2.$$

Then, the tensors $\mathcal{L}$ and $\mathcal{U}$ defining the box Ω_2 are given by

$$\mathcal{L} = \max\left\{\mathcal{M}_{\mathcal{B}} - \frac{\beta}{\max(\mathcal{V}_{\mathcal{B}})}\mathcal{V}_{\mathcal{B}}, 0\right\}, \quad and \quad \mathcal{U} = \min\left\{\mathcal{M}_{\mathcal{B}} + \frac{\beta}{\max(\mathcal{V}_{\mathcal{B}})}\mathcal{V}_{\mathcal{B}}, 1\right\},$$

where β is a positive constant which controls the tightness of the bounds, we set $\beta = 15$.

Example 01 : In the first example, we show the effectiveness of the proposed approach to restore "hibiscus.bmp"original color image of size $512\times 512\times 3$ that have

Figure 8.4: Original image (left), blurred and noisy image with $SNR = 2.06$ (right).

been contaminated by a Gaussian uniform blur with $\sigma_1 = 6$, $\sigma_2 = 5$ and $d_1 = d_2 = 6$. An additive Gaussian noise of level $\nu = 10^{-1}$, was added to produce the blurred and noisy image $\mathcal{B}$ with $SNR = 2.06$ as shown in Figure 8.4. We set the tolerance $tol = 10^{-3}$ and the maximum number of iterations equal to 50.

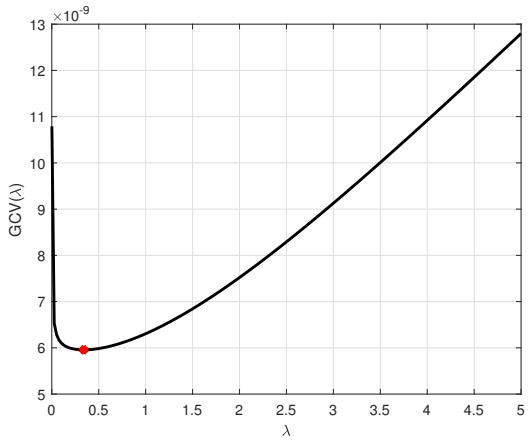


Figure 8.5: The *t-GCV* curve with the optimal regularization parameter value $\lambda^* = 3.425 \times 10^{-1}$.

Using the *t-GCV* method, the computed optimal regularization parameter is $\lambda^* = 3.425 \times 10^{-1}$ as illustrated in the curve of Figure 8.5. The restored images shown in Figure 8.6 are determined by solving the optimization problem (5.3) under two different constraint $\Omega_1 = B(\mathcal{L},\mathcal{U})$ and $\Omega_2 = \tilde{B}_r$ using Algorithm 5.2. The curves corresponding to the relative error (*RE*) and the *SNR* improvement, are illustrated in Figure 8.7. The curves corresponding to the *SNR* improvement and the relative error in Figure 8.7, and the results in Table 8.1, show that our algorithm using the constraint Ω_1 achieves the

Figure 8.6: Restored image using the constraint $\bar{B}_r$ with SNR = 17.23 (left), restored image using the constraint $B(\mathcal{L},\mathcal{U})$ with SNR = 15.53 (right).

best SNR with fast convergence.

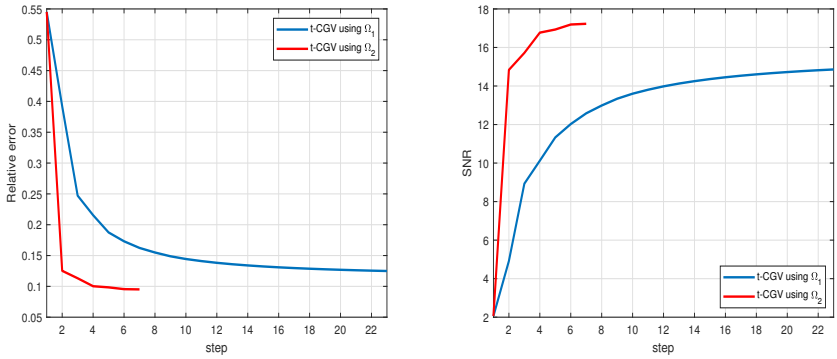


Figure 8.7: Relative error and SNR improvement curves.

Table 8.1: *t-CGT* approach using different constraints to recover "hibiscus.bmp"for different noise level.

ν	Ω	λ	$SNR(\mathcal{B})$	$SNR(\mathcal{X}^*)$	RE	Step	Time(s)
10^{-3}	$B(\mathcal{L},\mathcal{U})$	$7.5e^{-3}$	6.49	18.6	$8.12e^{-2}$	65	18.15
	$\bar{B}_r$			19.10	$7.67e^{-2}$	8	2.02
10^{-2}	$B(\mathcal{L},\mathcal{U})$	$5.57e^{-2}$	4.56	17.13	$9.61e^{-2}$	21	11.01
	$\bar{B}_r$			18.11	$8.59e^{-2}$	4	1.41
10^{-1}	$B(\mathcal{L},\mathcal{U})$	$3.425e^{-1}$	2.06	15.53	$1.15e^{-1}$	26	11.98
	$\bar{B}_r$			17.23	$9.51e^{-2}$	6	1.64

Example 2 : In the second example, we compare the proposed method (*t-CGT*) using Ω_1, with the classical method based on repeating the grayscale process for each channel under the same conditions, in order to recover the "peppers.png "color image of size $384 \times 512 \times 3$ from a Gaussian blur and Gaussian noise. For this example, we let $\sigma_1 = 4$, $\sigma_2 = 6$, $d_1 = d_2 = 6$ for the blurring operation, and we added a Gaussian noise with level $\nu = 2 \times 10^{-1}$. The original and the contaminated images are shown in Figure 8.8. The criterion for stopping the two algorithms consists of the tolerance $tol = 10^{-2}$ and the maximum number of iterations equal to 50.

Figure 8.8: Original image (left), blurred and noisy image with $SNR = 1.64$ (right).

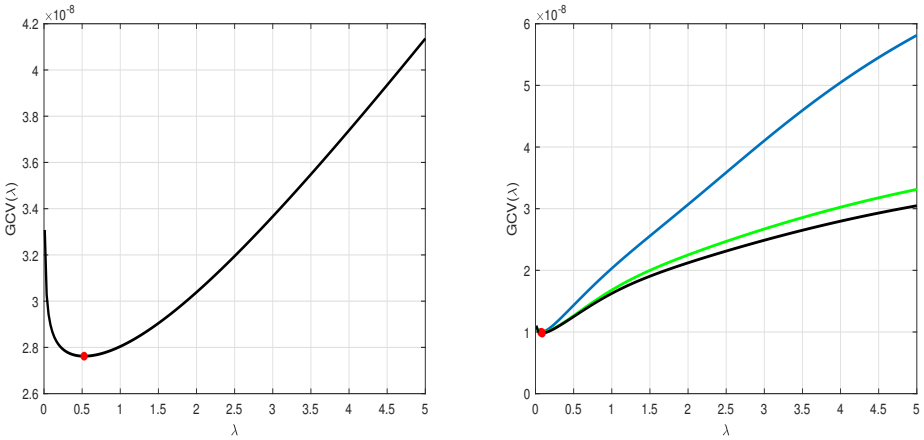


Figure 8.9: The *t-GCV* curve associated with the proposed method with the optimal value $\lambda^* = 5.29 \times 10^{-1}$ (left). The *GCV* curves associated to the three channels with the optimal values equal to $\lambda_1^* = 7.12 \times 10^{-2}$, $\lambda_2^* = 8.26 \times 10^{-2}$ and $\lambda_3^* = 8.83 \times 10^{-2}$, respectively (right).

Using the classical method, we compute the regularization parameter associated with each channel, as illustrated in Figure 8.9 (right). On the other hand, the proposed

method is based on the generalize version of *GCV* using tensor algebra leads to one regularization parameter as illustrate if Figure 8.9 (left). The fact of treating this problem of restoration by using tensor algebra, allows us to generalize the model of restoration in tensor format, and thus the function *t-GCV* for the regularization parameter selection associated with this model. The first line of Table 8.2 illustrates the comparison results obtained for the classical and the *t-CGT* methods to recover "peppers.png "color image of size $384 \times 512 \times 3$.

Figure 8.10: Restored image using the proposed algorithm with $SNR = 14.60$ (left), restored image using the classical approach with $SNR = 10.1$ (right).

In addition, Table 8.2 shows the comparison results for "car.avi"grayscale video of size $360 \times 640 \times 100$, and for "xylophone.mpg"grayscale video of size $240 \times 320 \times 140$. We report the *SNR* value, the relative error and the *CPU* time. It can be seen that for different tensor data the proposed method is more efficient and faster than the classical method, it achieves the best result in term of *SNR*, relative error, and CPU time. Furthermore, in the case of video restoration, the *t-GCV* developed required the computation of one regularization parameter instead of computing all the regularization parameters associated to all the frames of the video.

Table 8.2: Comparison results of *t-CGT* proposed algorithm and the classical approach.

Data	Method	$SNR(\mathcal{B})$	$SNR(\mathcal{X}^*)$	RE	iter	Time(s)
"peppers.png "	*t-CGT*	1.64	14.6	$1.11e^{-1}$	8	1.53
	Classical	1.64	10.1	$2.99e^{-1}$	12	2.08
"car.avi "	*t-CGT*	3.07	10.33	$1.21e^{-1}$	8	34.20
	Classical	3.07	9.73	$1.30e^{-1}$	10	71.96
"xylophone.mpg "	*t-CGT*	2.29	12.26	$9.89e^{-2}$	8	15.71
	Classical	2.29	11.78	$1.04e^{-1}$	12	21.40

Example 3 : In the present example, we are interested to restore grayscale video using Algorithm 5.2 with the constrained Ω_1. We consider "Chaplin.avi"video of size $360 \times 480 \times 200$. We set $\sigma_1 = 5$ and $\sigma_2 = 3$ for H_1 and H_2 respectively, and $d_1 = d_2 = 6$ and adding a white Gaussian noise of level $\nu = 10^{-2}$. In the top of Figure 8.11, we displayed the 11^{st}, 76^{th} and 143^{th} exacts (original) frames and the contaminated version, respectively. The *t-GCV* produce an optimal regularization parameter equal to $\lambda = 4.01 \times 10^{-2}$. The frames in the bottom of Figure 8.11 represent a sample from the recovered video.

Figure 8.11: Original frames (top), blurred and noisy frames with $SNR = 4.85$ (center) and restored frames with $SNR = 15.73$ in 92 s (bottom).

8.2.2 Tensorial total variation regularization algorithms

As mentioned before, in addition to recovering the signal, the total variation regularization also recover the boundaries. Now, let us solve the tensorial ill-posed degradation problem (8.12) using the tensorial total variation algorithms developed in Chapter 6.

Denoising color image and video examples

We show the effectiveness of the TTV_1/L^2-$LSQR$ Algorithm 6.5 to restore "Lena.bmp"color image of size $256 \times 256 \times 3$ and the grayscale video "car.avi"of size $360 \times 440 \times 20$ that have been contaminated by an additive Gaussian noise of different level. Figure 8.12 illustrates the performance of Algorithm 6.5 in recovering "Lena.bmp "color image from an additive Gaussian noise of level $\nu = 10^{-2}$. After 5 steps, the computed optimal value for the regularization parameter was $\mu_5 = 1.029e^{-1}$.

(a) Original image. (b) Noisy image (SNR = 7.44). (c) Restored image(SNR = 13.52).

Figure 8.12: Denoising "Lena.bmp" color image from an additive Gaussian noise.

In Figure 8.13, we present the restoration obtained with the same Algorithm 6.5 to denoise "car.avi "grayscale video from a Gaussian noise of level $\nu = 10^{-2}$. The discrepancy principle produced a regularization parameter given by $\mu_4 = 1.071e^{-1}$. Table 8.3 compares the computing time (in seconds), the relative errors, the SNR of the

(a) Original 1^{st} frame. (b) Noisy 1^{st} frame (SNR = 9.04). (c) Restored 1^{st} frame(SNR = 17.40).

Figure 8.13: Denoise "car.avi" grayscale video from a Gaussian noise.

computed restorations and the number of steps for different type of data and different noise level.

Table 8.3: Comparison results of Algorithm 6.5 for different noise level.

Data	*noise level*	*SNR*	*RE*	*Steps*	*CPU time (s)*
Lena.bmp	10^{-3}	20.45	$3.93e^{-2}$	5	0.59
	10^{-2}	13.52	$8.72e^{-2}$	5	0.67
car.avi	10^{-3}	23.96	$2.75e^{-3}$	5	9.8
	10^{-2}	17.41	$5.86e^{-2}$	5	9.9

Deblurring and denoising color image and video examples

First, let us compare the performance of Algorithms 6.1, 6.2, 6.3, and 6.4 to recover four color images: the "butterfly.tif "and "house.bmp "color images of size $256 \times 256 \times 3$, the "flowers.jpg "color image of size $340 \times 560 \times 3$, and the "hibiscus.bmp "color image of size $512 \times 512 \times 3$. All experimental color images are shown in Figure 8.14.

(a) *butterfly.tif* (b) *house.bmp* (c) *hibiscus.bmp*

(d) *flowers.jpg*

Figure 8.14: Experimental test color images.

A Gaussian blur and the salt and pepper noise of different density have been used to corrupt those images. We choose the parameters $\mu = 12$, $\beta = 80$, and $\rho = 4$. Also, we take $tol = 5 \times 10^{-3}$. The four approaches achieve a remarkable *SNR* improvement,

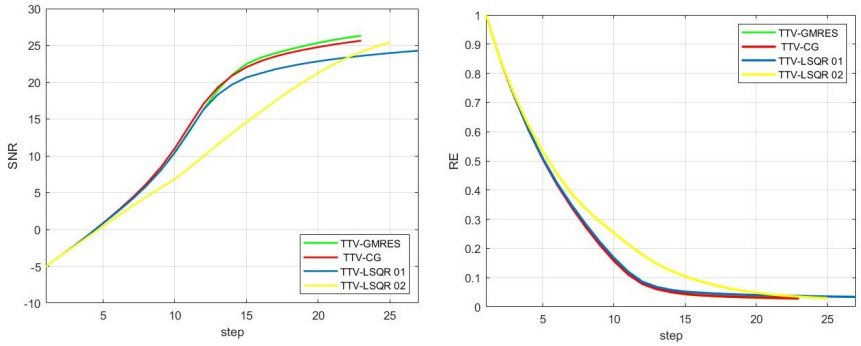
as illustrate in Figure 8.15 and the curves of Figure 8.16 for "flowers.jpg "color image. Table 8.4 reports results of the performance of the proposed TTV model for different projection methods, with different tensor data, which shows that the TTV using the gradient method is faster in terms of execution time than any other algorithm.

(a) Original image. (b) Corrupted image.

(c) Restored image with TTV_2/L^1-$GMRES$ Algorithm 6.1. (d) Restored image with TTV_2/L^1-CG Algorithm 6.2.

(e) Restored image with TTV_2/L^1-$LSQR$ 01 Algorithm 6.3. (f) Restored image with TTV_2/L^1-$LSQR$ 02 Algorithm 6.4.

Figure 8.15: Tensorial total variation results using optimized projection methods to restore "flowers.jpg"color image.



(a) The *SNR* improvement. (b) The relative error.

Figure 8.16: Convergence plots of the *SNR* improvement and the relative error for the tensorial total variation algorithm to restore "flowers.jpg"color image.

Table 8.4: Comparison between the four algorithms to restore four color images.

Data/size	Initial SNR	Methods	SNR	*ER*	Steps	CPU time(s)
Butterfly.tif $256 \times 256 \times 3$	**5.84**	TTV_2/L^1-*GMRES* Alg 6.1	**22.04**	$4.20e^{-2}$	21	3.9
		TTV_2/L^1-*CG* Alg 6.2	21.76	$4.27e^{-2}$	22	**2.6**
		TTV_2/L^1-*LSQR* Alg 6.3	21.79	$4.25e^{-2}$	45	8.2
		TTV_2/L^1-*LSQR* Alg 6.4	20.31	$5.05e^{-2}$	16	7.0
House.tif $256 \times 256 \times 3$	**4.08**	TTV_2/L^1-*GMRES* Alg 6.1	**21.40**	$3.03e^{-2}$	18	2.7
		TTV_2/L^1-*CG* Alg 6.2	20.16	$3.42e^{-2}$	17	**1.8**
		TTV_2/L^1-*LSQR* Alg 6.3	20.61	$3.24e^{-2}$	33	5.7
		TTV_2/L^1-*LSQR* Alg 6.4	19.07	$3.88e^{-2}$	16	7.1
flowers.jpg $340 \times 560 \times 3$	**2.12**	TTV_2/L^1-*GMRES* Alg 6.1	**26.57**	$2.70e^{-2}$	23	11.9
		TTV_2/L^1-*CG* Alg 6.2	25.65	$2.92e^{-2}$	23	**8.1**
		TTV_2/L^1-*LSQR* Alg 6.3	24.28	$3.42e^{-2}$	27	12.9
		TTV_2/L^1-*LSQR* Alg 6.4	25.40	$3.01e^{-2}$	25	36.1
hibiscus.bmp $512 \times 512 \times 3$	**1.61**	TTV_2/L^1-*GMRES* Alg 6.1	23.44	$4.75e^{-2}$	29	15.3
		TTV_2/L^1-*CG* Alg 6.2	22.19	$5.37e^{-2}$	31	**10.9**
		TTV_2/L^1-*LSQR* Alg 6.3	**24.20**	$4.26e^{-2}$	33	19.5
		TTV_2/L^1-*LSQR* Alg 6.4	16.31	$1.05e^{-1}$	63	87.1

Table 8.5: Restoration of grayscale videos.

Grayscale Video	*Initial SNR*	*Method*	*SNR*	*ER*	*Steps*	*CPU time(s)*
car-phone.avi $144 \times 176 \times 20$	**2.60**	*Standard matrix TV* [14]	16.78	$8.17e^{-2}$	12	6.07
		Reduced TTV algorithm	**17.18**	$7.70e^{-2}$	13	**4.17**
news.avi $144 \times 176 \times 60$	**1.31**	*Standard matrix TV* [14]	16.42	$7.92e^{-2}$	27	41.60
		Reduced TTV algorithm	**17.18**	$7.30e^{-2}$	16	**15.25**
car.avi $360 \times 360 \times 40$	**1.45**	*Standard matrix TV* [14]	18.18	$5.69e^{-2}$	11	34.95
		Reduced TTV algorithm	**20.44**	$4.34e^{-2}$	11	**25.44**

On the other hand, Table 8.5 illustrates a comparison between *TTV* approach and the standard form of *TV* using the matrix version [14] apply to restore blurred and

noisy grayscale videos corrupted using Gaussian blur with $\sigma_1 = 2$, $\sigma_2 = 1$, and $b = 4$ and different salt-and-pepper noise. The criterion for stopping the two algorithms consists of the tolerance $tol = 10^{-2}$ and the maximum number of iterations equal to 100.

(a) Original **first** frame

(b) Blurred and noisy frame.

(c) Restored frame using matrix version of *TV*.

(d) Restored frame using *TTV*.

(e) Restored frame using reduced *TTV*.

Figure 8.17: Comparison of the tensorial *TV* algorithm and the matrix based algorithm to recover "car.avi"grayscale video from blur an noise.

By adopting the truncation parameters $r_1 = floor(\frac{2}{3}I_1)$ and $r_2 = floor(\frac{2}{3}I_2)$, with $I_1 \times I_2 \times I_3$ is the size of the experimental data, the reduced *TTV* algorithm, discussed in Section 6.6, achieves the best *SNR* improvement in less time than the matrix version. The frame comparison is given in Figure 8.17 for the "car.avi"grayscale video of size $360 \times 440 \times 20$. Finally, the reduced *TTV* applying on a color video of size $360 \times 640 \times 3 \times 40$ degraded by Gaussian blur and salt and pepper noise, give us Figure 8.18 in 4.89 min.

Comparisons with some state-of-art method

In this section, we compare numerically, under the same conditions, our algorithm referred to as the *TTV* algorithm with four state-of-art algorithms developed for total variation regularization based image restoration. The derivative alternating direction method of multipliers (*D-ADMM*) algorithms, referred to as *D-ADMM-C* for conventional *D-ADMM* algorithm and *D-ADMM-H* for hybrid *D-ADMM* algorithm [130], the

Figure 8.18: Frames of the original video (top), frames of the blurred and noisy video (center), frames of the restored video(bottom).

block-matching and 3D filtering, referred to as *BM3D-DEB* algorithm [39] that often produce state-of-the-art image deblurring results, and the joint statistical modeling in space-transform domain [152], referred to as *JSM*.

For comparing the accuracy obtained by our proposed algorithm with those of the four algorithms, we have used the color image *"peppers.bmp"* size $256 \times 256 \times 3$. We generated a blurred and noisy image from a true image by using three blur kernels, a uniform kernel, a unsharp kernel and a Gaussian blur kernel, and two different noise, additive Gaussian noise and salt and pepper (see Table 8.6). In order to provide a fair and unified framework for comparison, all these algorithms are endowed with the same convergence criterion, i.e., the iterations for all algorithms were terminated when the relative error between two successive iterates of approximated primal variable is less than the tolerance $tol = 10^{-3}$ or when a maximum of 500 iterations has been performed.

In Table 8.6, for each algorithm, we report the relative error, the *SNR*, as well as the *CPU*-time in seconds. We can observe that the proposed *TTV* algorithm generally can achieve better restoration quality than the other algorithms in terms of both *SNR* and

Table 8.6: The computational results for the *D-ADMM-H*, *D-ADMM-C*, *BM3D-DEB*, *JSM*, and *TTV* algorithms to recover *"peppers.bmp "*color image of size $260 \times 240 \times 3$ from different blur and noise choices.

Noise	*Level*	*Blur*	*Initial SNR*	*Method*	*SNR*	*RE*	*CPU time (s)*
Gaussian noise	$1e^{-3}$	*Gaussian blur*	14.31	*D-ADMM-H*	19.72	$6.1e^{-2}$	0.81
				D-ADMM-C	20.37	$5.7e^{-2}$	1.45
				BM3D-DEB	22.75	$4.3e^{-2}$	11.86
				JSM	23.15	$4.1e^{-2}$	68.12
				Proposed TTV	**23.30**	$4.1e^{-2}$	17.03
	$1e^{-1}$	*Unsharp blur*	-8.34	*D-ADMM-H*	–7.27	1.37	0.19
				D-ADMM-C	–7.27	1.36	0.24
				BM3D-DEB	20.85	$5.4e^{-2}$	11.90
				JSM	21.51	$5e^{-2}$	68.19
				Proposed TTV	**21.50**	$5e^{-2}$	1.95
Salt and pepper noise	0.1	*Gaussian blur*	3.18	*D-ADMM-H*	5.23	$3.25e^{-1}$	1.81
				D-ADMM-C	6.35	$2.85e^{-1}$	4.32
				BM3D-DEB	12.06	$1.48e^{-1}$	12.01
				Proposed TTV	**24.02**	$3.72e^{-2}$	11.41
	0.2	*Uniform blur*	0.32	*D-ADMM-H*	1.39	$5.05e^{-1}$	1.34
				D-ADMM-C	1.23	$5.14e^{-1}$	2.26
				BM3D-DEB	4.76	$3.43e^{-1}$	11.27
				Proposed TTV	**22.09**	$4.70e^{-2}$	18.28

RE values. When the noise level is low, the *TTV* algorithm is comparable with the *BM3D* algorithm in terms of both *SNR* and *RE* values but, in this experiences, we observed that the *TTV* converges much faster than *BM3D*. When the noise level is high, especially using salt and pepper noise, our proposed algorithm performs better.

8.3 Color image and video completion

Completion is a technique of filling missing elements of incomplete data using the values of available elements and the structural assumptions of data. Let us consider the Nth-order tensor $\mathcal{G}$ with observed entries indexed by the set C i.e. $C = \{(i_1, i_2, \ldots, i_N) : \mathcal{X}_{i_1,i_2,\ldots,i_N}$ is observed $\}$. Following the same definition of Candes and Tao in [27], we define, in the tensor form, the projection $P_C(\mathcal{X})$ to be the Nth-order tensor with the observed elements of $\mathcal{X}$ preserved and the missing entries replaced with 0, namely,

$$P_C(\mathcal{X}) = \begin{cases} \mathcal{X}_{i_1,i_2,\ldots,i_N} & if\ (i_1, i_2, \ldots, i_N) \in C, \\ 0 & otherwise. \end{cases} \tag{8.13}$$

One of the variants of the data completion problem is to find the lowest rank which matches the observed data. This leads to an NP-hard rank minimization problem due to the non-convexity of the rank function [143]. For that purpose, we need to replace the rank function with a similar expression that provides the same results. Therefore, the

nuclear norm minimization method [111, 52, 153] is widely used in this case to replace the rank minimization problem by the following one

$$\begin{cases} \min\limits_{\mathcal{X}} \|\mathcal{X}\|_* \\ \textit{subject to} \;\; P_C(\mathcal{X}) = P_C(\mathcal{G}). \end{cases} \tag{8.14}$$

where $\|\,.\,\|_*$ stands for the tensor nuclear norm defined as sum of the singular values of the n-mode matricization $\mathcal{X}_{(n)}$ of the tensor $\mathcal{X}$, i.e. $\|\mathcal{X}\|_* = \sum\limits_n \|\mathcal{X}_{(n)}\|_*$. Tensor completion via total variation minimization was proposed in [150, 108] as an efficient technique to regularize the minimization problem (8.14). The tensor total variation completion problem is given in the following form

$$\min_{\mathcal{X}} \|P_C(\mathcal{X}) - P_C(\mathcal{G})\|_F^2 + \|\mathcal{X}\|_* + \mu|||\nabla\mathcal{X}|||_1. \tag{8.15}$$

The problem (8.15) can be formulated to a constrained minimization problem as

$$\min_{\mathcal{X}\in\Omega} \|P_C(\mathcal{X}) - P_C(\mathcal{G})\|_F^2 + \mu|||\nabla\mathcal{X}|||_1, \tag{8.16}$$

where $\Omega = \{\mathcal{X},\ \|\mathcal{X}\|_* \leqslant \epsilon\}$.

The completion problem can be seen from different points of view depending on the algorithm that we will used. In the next section we will see different consideration to treat this problem using the accelerated double tensorial proximal gradient algorithms and the alternating tensorial conditional gradient algorithm.

8.3.1 Accelerated tensorial double proximal gradient algorithm

By setting $\mathscr{H} = P_C$ and $\mathcal{B} = P_C(\mathcal{G})$, the problem (8.16) leads to the main problem (7.1). Then, to solve the minimization problem (8.16), we can use our proposed accelerated tensorial double proximal gradient algorithm. Before applying the algorithm, let first compute the proximal mapping of the tensor nuclear norm.

Definition 8.3.1. *[97] The nuclear norm of $\mathcal{X}$ is defined using t-SVD decomposition $\mathcal{X} = \mathcal{U} *_t \mathcal{S} *_t \mathcal{V}^*$ as follow:*

$$\|\mathcal{X}\|_* = \sum_{i=1}^{r} \mathcal{S}(i,i,1), \tag{8.17}$$

where r denotes the tubal rank of $\mathcal{X}$.

Let Π_Ω denotes the projection on the convex set Ω. It is immediate to proof that the expression of the projection Π_Ω on Ω reduces to the proximal mapping of the nuclear norm. Namely, for all $\mathcal{Z}$ in the tensor space $\mathbb{T}_N$

$$\Pi_\Omega(\mathcal{Z}) = prox_{\sigma\|\,.\,\|_*}(\mathcal{Z}), \tag{8.18}$$

where the proximal mapping of the nuclear norm is given in Proposition 8.3.1.

Proposition 8.3.1. *[114] Let $\mathcal{U} *_t \mathcal{S} *_t \mathcal{V}^*$ be the t-SVD decomposition of the tensor $\mathcal{X}$. The proximal mapping of the nuclear norm is given by*

$$prox_{\sigma\|.\|_*}(\mathcal{X}) = \mathcal{U} *_t \mathcal{S}_\sigma *_t \mathcal{V}^*, \quad \text{for any } \sigma > 0, \tag{8.19}$$

where $\mathcal{S}_\sigma$ is the result of the inverse discrete Fourier transform (IDFT) on $max(\bar{\mathcal{S}} - \sigma, 0)$ along the third dimension, which means performing the IDFT on all the tubes. $\bar{\mathcal{S}}$ is the result of DFT on $\mathcal{S}$ along the third dimension.

In the following, we will be interested in two essential parts of tensor completion: Color image and video inpainting (text removal) and grayscale video completion. We first illustrate the performance of the proposed algorithms 7.3 and 7.6 by comparing three acceleration methods in inpainting different color images and grayscale videos. The completion of the grayscale video is reported after to show the efficiency of our algorithm in case of uniformly random missing pixels. Finally, we end with some comparisons with state-of-art algorithms.

Text removal

As an interesting application of completion problems, the text removal is a process of data inpainting that based on the completion techniques to recover the missing region in the tensor data or removing some objects added to it. The operation of inpainting depends on the type damaging in the image, and the application that caused this distortion. For example, in the text removal process, we talk about removing the text that can be found in an image or a video [140]. In the literature, many techniques have been developed to solve this problem [101, 55, 117]. The total variation was among the most efficient method to solve such a problem. In this example, we took the original Barbara color image of size $256 \times 256 \times 3$ and we added a text to this image as shown in Figure 8.19. The criterion for stopping the proposed algorithms consists of the tolerance

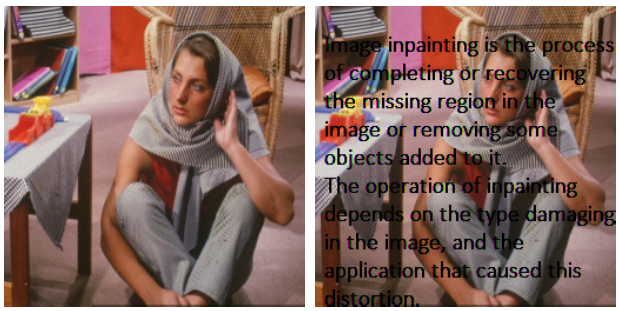


Figure 8.19: Original image (left), the observed image ($PSNR = 15.5$) (right).

$tol = 10^{-2}$, the maximum number of iterations $k_{max} = 200$. We hand turned all the

parameters λ, μ, α and σ by choosing each one in its appropriate interval. We hand turned the value of λ in the interval $(0,\ 1/L(\nabla P_\Omega)) = (0,\ 1/2)$ by choosing $\lambda = 2.5 \times 10^{-1}$. On the other hand the step size sequence(α_k) computed using the line search iterative

Figure 8.20: Inpainted image without acceleration ($PSNR = 21.71$), inpainted image with ATDPG-Nesterov ($PSNR = 32.45$), inpainted image with ATDPG-MPE ($PSNR = 32.5$), inpainted image with ATDPG-RRE ($PSNR = 32.47$).

method starting by $\alpha_0 = 1.1$ with a line search parameter $\tau = 0.5$. We set $\sigma = 6.5 \times 10^{-2}$ and finally the regularization parameter was chosen to be $\mu = 1.2e^{-2}$. The corresponding results are shown in Figure 8.20. Clearly, the accelerated version of the tensorial double

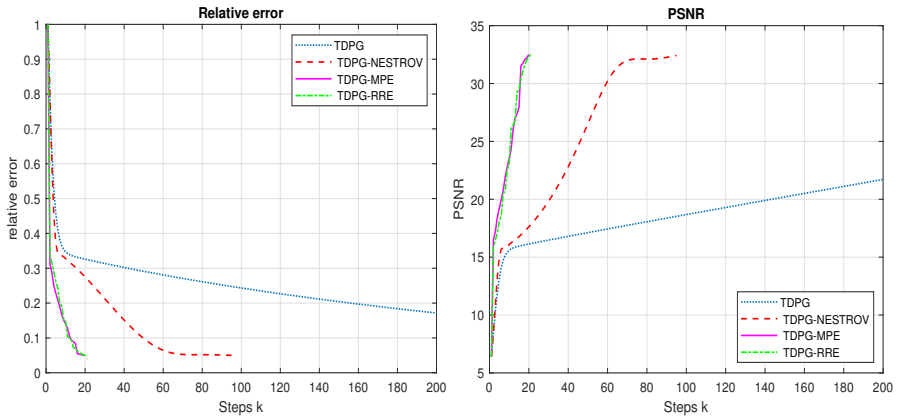


Figure 8.21: The PSNR and relative error curves.

proximal gradient method provide clearer images by removing all the added text, either using Algorithm 7.3 based on Nesterov acceleration approach or Algorithm 7.6 using the polynomial extrapolation techniques. Moreover, the relative error and the *PSNR* curves represented in Figure 8.21 show that the results produced by the *ATDPG* algorithm accelerated by the polynomial extrapolation techniques *RRE* or *MPE* converge faster than those produced by the *ATDPG* accelerated by Nesterov's technique. The speed of the convergence of the polynomial extrapolation method in comparison with the Nesterov's approach are clearly illustrated in the report of acceleration in Figure 8.22, that show the fast convergence of *ATDPG-MPE* and *ATDPG-RRE* in less iterations than *ATDPG-Nesterov*.

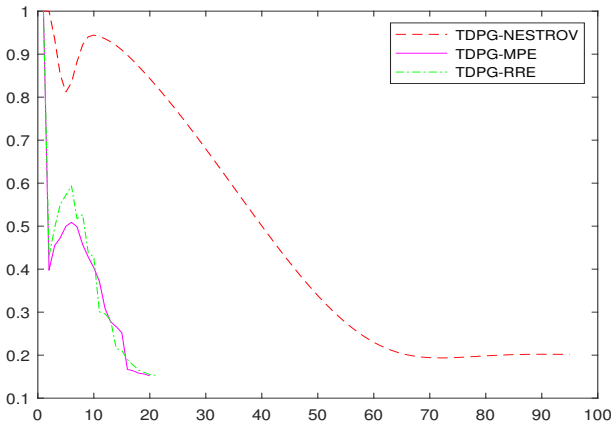


Figure 8.22: The report of acceleration $\frac{\|\mathcal{T}_k - \mathcal{X}_*\|}{\|\mathcal{X}_k - \mathcal{X}_*\|}$.

Table 8.7: Comparison between the proposed acceleration techniques for tensorial total variation proximal gradient method.

Data size	Algorithm	$PSNR(\mathcal{X}^*)$	RE	Step	time (s)
barbara.bmp $256 \times 256 \times 3$	*ATDPG*	21.71	$1.72e{-1}$	200	27.41
	ATDPG-Nesterov	32.45	$5e{-2}$	97	13.37
	ATDPG-MPE $(q = 5)$	32.50	$5e{-2}$	20	19.72
	ATDPG-RRE $(q = 5)$	32.47	$5e{-2}$	21	19.88
xylophone.mpg $120 \times 160 \times 30$	*ATDPG*	28.62	$4.80e{-2}$	18	6.71
	ATDPG-Nesterov	29.65	$3.98e{-2}$	12	4.10
	ATDPG-MPE $(q = 5)$	32.27	$2.99e{-2}$	3	5.38
	ATDPG-RRE $(q = 5)$	29.53	$4.03e{-2}$	3	5.66

In Table 8.7, we have reported the *PSNR* of the completed tensor, the relative error, as well as the number of iterations and the CPU-time results for "barbara.bmp "color image and "xylophone.mpg "grayscale video of size $120 \times 160 \times 30$. Based on the tests reported in Table 8.7 and many more unreported tests, we remark that our proposed algorithm works very effectively for image and video inpainting problems, in terms of the *PSNR* as well as in terms of the relative error.

Grayscale video completion

In order to have more quantitative evaluations on the proposed approach, we used the "news.mpg "grayscale video of size $144 \times 176 \times 10$ as original data and we randomly mask off about 80% of entries that we regard them as missing values, as shown in the second line of Figure 8.23.

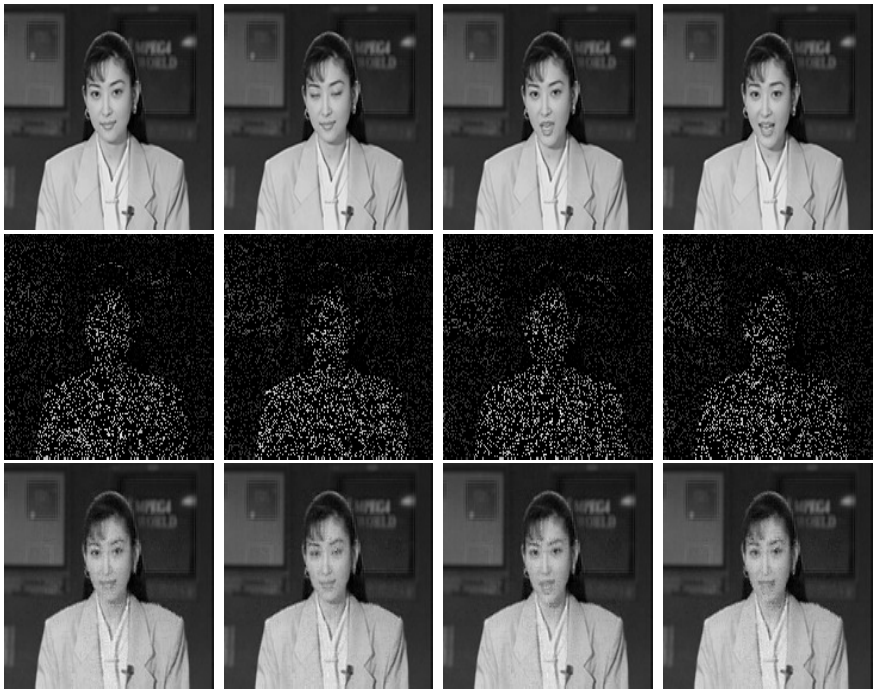


Figure 8.23: Original frames (top), incompleted frames with $PSNR = 7.94$ (center), recovered frames with $PSNR = 33.21$ (bottom).

The completed frames in Figure 8.23 with $PSNR = 33.21$ are obtained by using Algorithm 7.6 with the *GT-RRE* polynomial extrapolation technique. The criterion for stopping the algorithm consists of the tolerance $tol = 10^{-2}$ and the maximum number of iterations $k_{max} = 200$. We set the step size parameters $\lambda = 2 \times 10^{-1}$, $\alpha_0 = 1.1$, $\sigma = 6 \times 10^{-1}$ and the regularization parameter $\mu = 2 \times 10^{-2}$. We can see that the results are visually pleasant using the accelerated version of the tensorial double proximal gradient method that achieves a *PSNR* value equal to 33.21.

Comparison with some state-of-the-art algorithms

In this subsection, we compare the performance of our proposed method with the following state-of-the-art tensor completion algorithms: *LRTC*, *TNN*, *FBCP* and *RTC*. The *LRTC* algorithms are based on minimizing the sum of nuclear norms of the unfolded matrices of a given tensor. The *LRTC* has two versions, *HaLRTC* and *FaLRTC* [111]. The first one stands for a fast low-rank tensor completion algorithm and the second stands for a high accuracy low-rank tensor completion algorithm. We also found the *TNN* method [154] which is a tensor nuclear norm-based method developed using the tensor-Singular Value Decomposition (t-*SVD*) [96]. The concept of automatic tensor rank determination was introduced in [155] which is based on a Bayesian *CP* Factorization (*FBCP*) in order to recover incomplete tensor. For the same goal of completing tensors,

recently, the *RTC* algorithm [33] was developed as an auto-weighted approach using this time the well-known tensor trains decomposition [97].

Four benchmark color images, of size $256 \times 256 \times 3$, are used in the comparisons, Baboon, Lena, Flower, and Airplane (see Figure 8.24). To show the efficiency of the

Figure 8.24: Four benchmark color images: Baboon, Lena, Flower and Airplane, respectively.

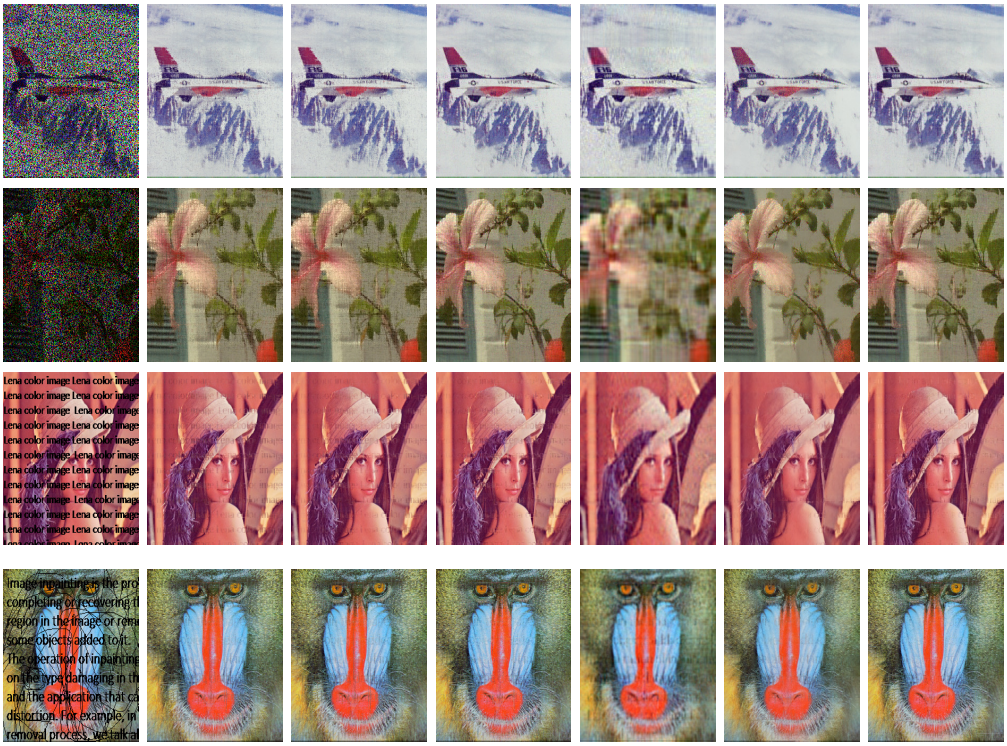
proposed algorithm for different types of tensor completion, we generate different incomplete images either using uniformly random missing pixels or non-random missing pixels. In the first case, 80% and 60% of missing pixels are uniformly distributed in Flower and Airplane color images, respectively. Non-random missing pixels, such as text and scrabble, are used to corrupt the color images of Lena and Baboon. The corrupted images are shown in the first column of Figure 8.25. In order to provide a fair and unified framework for comparison, all six algorithms are endowed with the same convergence criterion, i.e. the iterations for all algorithms were terminated when the relative error between two successive iterates of approximated primal variable is less than the tolerance $tol = 10^{-4}$ or when a maximum of 200 iterations has been performed. In addition, the parameters of the six algorithms are refined in relation to the best *PSNR*,

Table 8.8: Comparison of the results of six methods applied to four different images.

Data	Methods	*FaLRTV*	*HaLRTV*	*TNN*	*FBCP*	*RTC*	*ATDPG*
Airplane	*PSNR*	25.39	26.14	29.37	26.70	26.83	29.03
	RE	6.69e−2	6.13e−2	4.23e−2	5.75e−2	5.66e−2	4.40e−2
	Time (s)	89.44	58.76	26.30	38.43	15.64	36.44
Flower	*PSNR*	22.04	22.40	24.94	24.29	23.84	24.97
	RE	1.64e−1	1.58e−1	1.18e−1	1.27e−1	1.34e−1	1.17e−1
	Time (s)	214.71	117.12	43.81	83.66	43.70	28.05
Lena	*PSNR*	28.13	28.94	28.91	26.39	28.57	30.29
	RE	7.07e−2	6.44e−2	6.46e−2	8.64e−2	6.73e−2	5.51e−2
	Time (s)	226.40	148.06	42.67	27.19	8.81	44.91
Baboon	*PSNR*	25.69	25.45	25.41	21.13	25.10	26.84
	RE	9.63e−2	9.89e−2	9.94e−2	1.63e−1	1.03e−1	8.44e−2
	Time (s)	133.15	72.92	28.91	23.38	8.30	25.54

RE, and *CPU* times scores on the images. Table 8.8 reports the *PSNR*, the relative error

RE, as well as the *CPU* time in seconds for all the six algorithms. While the recovered images are shown in Figure 8.25.



(a) (b) (c) (d) (e) (f) (g)

Figure 8.25: Image completion comparisons of Airplane, Flower, Lena and Baboon by six algorithms. (a) the column of the observed (incomplete) images, (b) the completed images with *FaLRTV* algorithm, (c) the completed images with *HaLRTV* algorithm, (d) the completed images with *TNN* algorithm, (e) the completed images with *FBCP* algorithm, (f) the completed images with *RTC* algorithm and (g) the completed images with the proposed *ATDPG* algorithm.

For the uniformly random examples, the proposed *TGPG* algorithm is comparable with the TNN algorithm. Both approaches reach the best results in terms of *PSNR* and *RE*. However, by increasing the missing pixels in the Flower color image, the proposed algorithm converges faster than *TNN*. On the other hand, in the non-random example, the proposed algorithm achieves the best *PSNR* and RE among all other methods. Meanwhile, in terms of computation time, algorithms *RTC* and *FBCP* are faster than ours.

8.3.2 Alternating tensorial conditional gradient algorithm

Another reformulation that can be adopted to represent the completion model is the following

$$\begin{cases} \min\limits_{\mathcal{X}} \|\mathcal{X}-\mathcal{G}\|_F^2, \\ s.t\ \mathcal{X}_{i_1,i_2,\ldots,i_N} = \mathcal{G}_{i_1,i_2,\ldots,i_N},\ \forall (i_1,i_2,\ldots,i_N) \in J, \end{cases} \tag{8.20}$$

where the Nth-order tensors $\mathcal{X}$ and $\mathcal{G}$ represent the approximate solution and the incomplete data, respectively, and the set J contains the observed positions such that

$$J = \{(i_1,i_2,\ldots,i_N) :\ \mathcal{X}_{i_1,i_2,\ldots,i_N}\ \textit{is observed}\ \}.$$

Thus, it is immediate to see that the problem (8.20) can be regularized using the tensorial total variation. As a consequence, the regularized problem may be written in the form of (7.2) by considering the convex $\mathcal{C}$ set as

$$\mathcal{C} = \{\mathcal{X} \in \mathbb{T}_N :\ \mathcal{X}_{i_1,i_2,\ldots,i_N} = \mathcal{G}_{i_1,i_2,\ldots,i_N},\ \forall (i_1,i_2,\ldots,i_N) \in J\}. \tag{8.21}$$

It is clear that the set $\mathcal{C}$ is a closed convex set in $\mathbb{T}_N$. Then, the projection on $\mathcal{C}$ is given by $P_{\mathcal{C}}(\mathcal{Y}) = \mathcal{Z}$ with

$$\mathcal{Z}_{i_1,i_2,\ldots,i_N} = \begin{cases} \mathcal{G}_{i_1,i_2,\ldots,i_N} & \textit{if}\ (i_1,i_2,\ldots,i_N) \in J, \\ \mathcal{Y}_{i_1,i_2,\ldots,i_N} & \textit{otherwise.} \end{cases} \tag{8.22}$$

Line removal

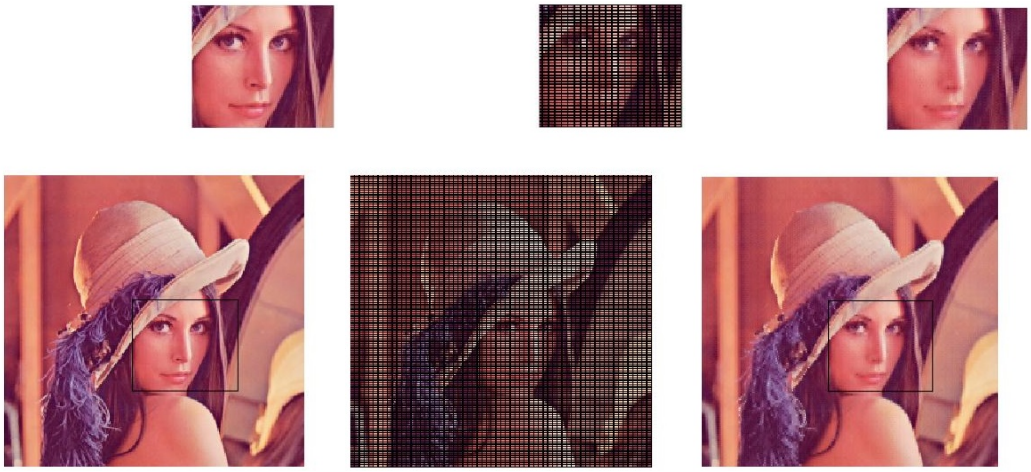


Figure 8.26: *"Lena"*original color image (left), the incomplete image with *PSNR* = 7.159 (center) and the restored image (right) with *PSNR*= 30.264.

In the example of benchmark *"Lena"*color image of size $256 \times 256 \times 3$, the missing entries are chosen as random black lines, see Figure 8.26. The observed missing color image have an initial *PSNR* equal to 7.159. Algorithm 7.2 consists of the tolerance $tol = 10^{-3}$, the maximum number of iterations $Itermax = 100$ and a regularization parameter $\mu = 10^{-3}$. The image completed by the proposed algorithm has an *PSNR* equal to 30.26 as shown in Figure 8.26. The results illustrated by Figure 8.26 show that the proposed algorithm not only completes the missing area in the color image, but also makes it as close as possible to the original images.

Grayscale video completion

In this example, we are interested in the completion of a grayscale video. We consider the *"xylophone"*video from MATLAB. The video clip is in .mp4 format, it is a grayscale video of 40 frames of 240×320 pixels, which can be seen as third-order tensor of size $240 \times 320 \times 40$. We mask off 70% of entries in the video randomly. The criterion for stopping the proposed algorithm consists of the tolerance was $tol = 10^{-2}$ and the maximum number of iterations $k_{max} = 50$. The regularization parameter was chosen to be $\mu = 4 \cdot 10^{-4}$. Figure 8.28 illustrates the efficiency of the proposed algorithm that presents a remarkable improvement from the corrupted into the recovered frames. Figure 8.27 shows the relative error and the *PSNR* plots for missing rates ranging from 10% to 90%, where the missing rate is defined to be the percentage of pixels which are unknown.

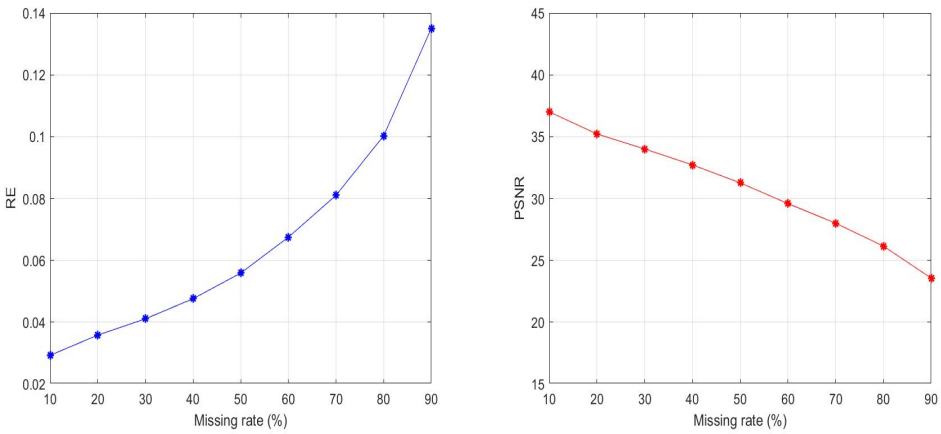


Figure 8.27: The relative error and the *PSNR* plots against missing rate for *"xylophone"*grayscale video.

From Figure 8.27, it seems that practically *ATCG-TV* is able to reach very remarkable accuracies and *PSNR* improvement for different missing rate values. We can remark that

even with 90% of missing data, the recovered video obtained by the proposed algorithm achieve an *PSNR* value equal to 24.

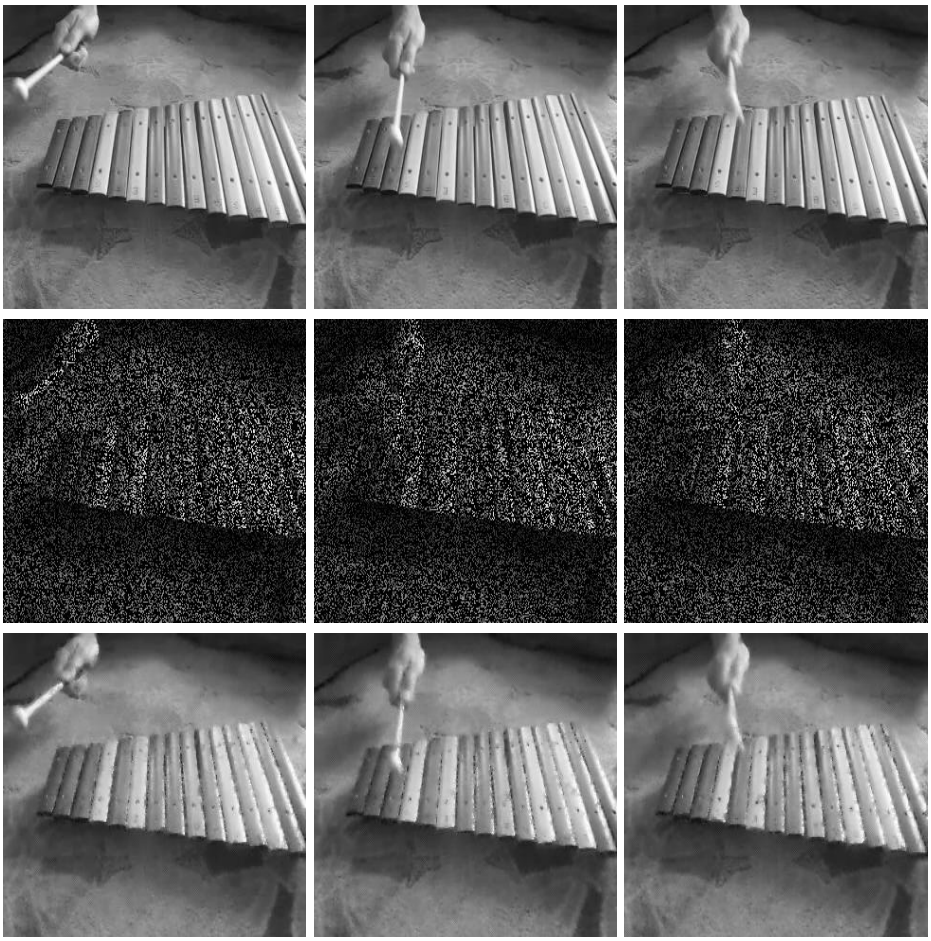


Figure 8.28: *"xylophone"*frames no 1, 20, and 40 from left to right respectively. Original frames (top), corrupted frames with 70% of the missing entries and $PSNR = 7.805$ (center) and the completed frames with $PSNR = 28.076$ (bottom).

Comparison with some state-of-the-art algorithms

In the last example, we will compare our proposed approach with some state of art methods for images and videos completion using tensor algebra such as *TNN* where the algorithm based on tensor-SVD [154], *LTRT* based on the low tubal rank tensor minimization [115] and *TCTF* algorithm [156]. For the comparison, we use the test color image *"barbara"*of size $256 \times 256 \times 3$, see Figure 8.29, we mask off between 50% to 99% to compare the efficiency of the four algorithms. All the algorithms are endowed with the

same convergence criterion, i.e., the tolerance was $tol = 10^{-2}$ and a maximum number of iterations equal to 300.

Figure 8.29: Original image (left), incomplete image with 90% missing rate (right).

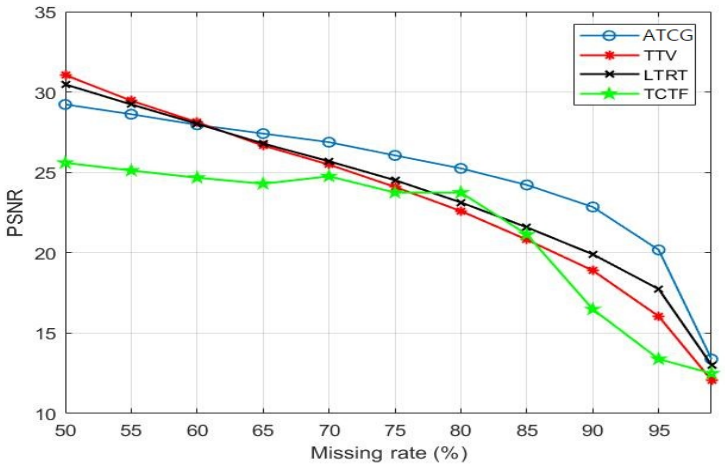
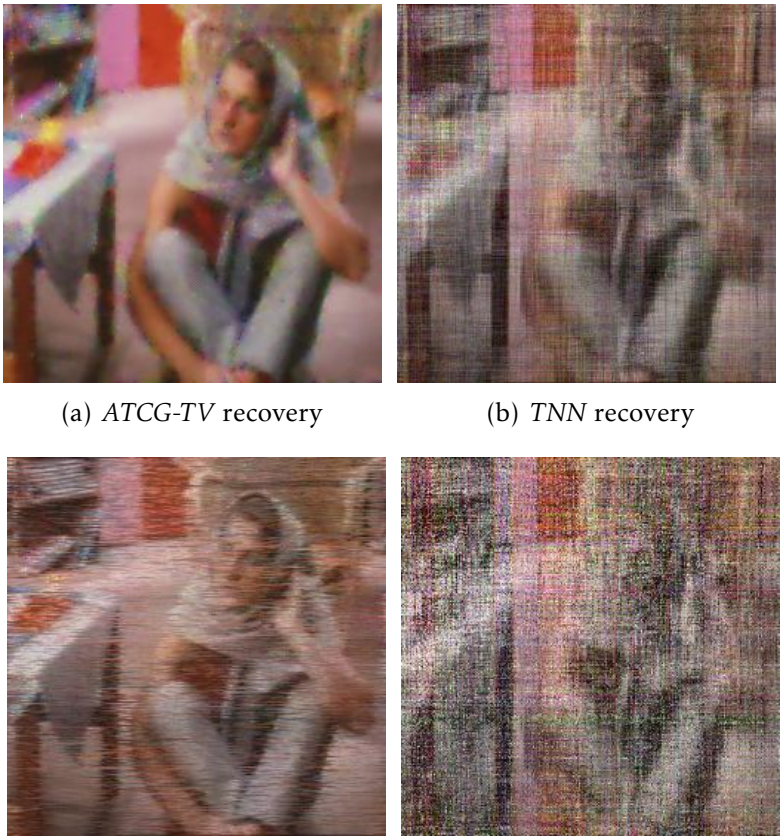


Figure 8.30: The *PSNR* plots against missing rate of the four algorithms for *"barbara"*color image.

In Figure 8.31, we plotted the *PSNR* for the missing rates ranging from 50% to 99%. When the missing rates are high, we can see that the proposed algorithm *ATCG-TV* achieves the best results. The same remark can be seen in Figure 8.30 that shows the performance of the four algorithms to complete 90% of *"barbara"*color image.



(a) *ATCG-TV* recovery (b) *TNN* recovery

(c) *LTRT* recovery (d) *TCTF* recovery

Figure 8.31: Results of the algorithms *ATCG-TV*, *TNN*, *LTRT* and *TCTF* for "*barbara*"color image.

8.4 Conclusion

The different numerical results provided in this chapter illustrate the efficiency of the developed tensorial regularization algorithm in the field of the color image and video processing. The examples we have reported are only part of a class of numerical tests on different cases and different parameters that we have investigated. However, the remaining point that is still under investigation is parameter selection. We only developed some parameter selection methods in particular cases, when the structure of the cost function is simple, such as in the case of Tikhonov regularization. But generally, we apply our algorithm with a range of different parameters and only report the best result for each regularizer in terms of *PSNR*/*SNR* and *RE*.

CHAPTER 9

General conclusions and future directions

Starting from the representation of higher dimensional inverse problems to the adaptation of some regularization techniques to solve these problems, this thesis has provided a study of several approaches and algorithms by introducing tools from multilinear algebra and convex optimization.

Tensor tools, such as the n-mode product and the t-product, have been used to model some high-dimensional inverse problems. Such a representation ensures a more general model that preserves the structure of the data as well as the accuracy of the solution. After the representation step, an adaptation of two regularization methods, Tikhonov and total variation, was provided in various settings to cover the largest number of inverse problems. Projection methods, extrapolation methods and other notions have played an essential role in the development of the proposed algorithms.

Performing a series of numerical tests confirmed the importance of selecting a suitable regularization method and a suitable regularization parameter. The most delicate points of the regularization techniques was the selection of parameters. Some methods require the intervention of some parameters that improve the quality of the resolution, however, the identification of these parameters is difficult and influences the convergence of the method.

Following the work done in this dissertation, we will be interested in developing new approaches for selecting the regularization parameter, especially in the case of non-quadric cost function problems such as the total variation. Far from the resolution process, the algebraic structure of the tensor space as a space of matrices over a commutative ring is a path that we still work on to generalize all the matrix notions.

Bibliography

[1] Yuri A Abramovich and Charalambos D Aliprantis. *An invitation to operator theory*, volume 50. American Mathematical Society Providence, RI, 2002.

[2] Hedy Attouch, Giuseppe Buttazzo, and Gérard Michaille. *Variational analysis in Sobolev and BV spaces: applications to PDEs and optimization*. SIAM, 2014.

[3] Heinz H Bauschke, Regina S Burachik, Patrick L Combettes, Veit Elser, D Russell Luke, and Henry Wolkowicz. *Fixed-point algorithms for inverse problems in science and engineering*, volume 49. Springer Science & Business Media, 2011.

[4] Heinz H Bauschke, Patrick L Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

[5] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[6] Fatemeh PA Beik, Khalide Jbilou, Mehdi Najafi-Kalyani, and Lothar Reichel. Golub–Kahan bidiagonalization for ill-conditioned tensor equations with applications. *Numerical Algorithms*, 84(4):1535–1563, 2020.

[7] Fatemeh Panjeh Ali Beik, Mehdi Najafi-Kalyani, and Lothar Reichel. Iterative Tikhonov regularization of tensor equations based on the Arnoldi process and some of its generalizations. *Applied Numerical Mathematics*, 151:425–447, 2020.

[8] O Benchettou, AH Bentbib, A Bouhamidi, and K Kreit. Tensorial conditional gradient method for solving multidimensional ill-posed problems. *Applied Numerical Mathematics*, 173:222–238, 2022.

[9] O. Benchettou, A.H. Bentbib, A. Bouhamidi, and K. Kreit. Tensorial conditional gradient method for solving multidimensional ill-posed problems. *Applied Numerical Mathematics*, 173:222–238, 2022. ISSN: 0168-9274. DOI: https://doi.org/10.1016/j.apnum.2021.12.002. URL: https://www.sciencedirect.com/science/article/pii/S0168927421003354.

[10] Oumaima Benchettou, Abdeslem Hafid Bentbib, and Abderrahman Bouhamidi. Tensorial total variation-based image and video restoration with optimized projection methods. *Optimization Methods and Software*:1–32, 2022.

[11] Abdeslem Hafid Bentbib, Abderrahman Bouhamidi, Karim Kreit, et al. A conditional gradient method for primal-dual total variation-based image denoising, 2018.

[12] Abdeslem Hafid Bentbib, Mohamed El Guide, Khalide Jbilou, and Lothar Reichel. A global lanczos method for image restoration. *Journal of Computational and Applied Mathematics*, 300:233–244, 2016.

[13] AH Bentbib, M El Guide, K Jbilou, E Onunwor, and L Reichel. Solution methods for linear discrete ill-posed problems for color image restoration. *BIT Numerical Mathematics*, 58(3):555–576, 2018.

[14] AH Bentbib, M El Guide, and Khalide Jbilou. A generalized matrix krylov subspace method for tv regularization. *Journal of Computational and Applied Mathematics*:112405, 2019.

[15] Åke Björck. Least squares methods. *Handbook of numerical analysis*, 1:465–652, 1990.

[16] A Bouhamidi and K Jbilou. Sylvester tikhonov-regularization methods in image restoration. *Journal of Computational and Applied Mathematics*, 206(1):86–98, 2007.

[17] A Bouhamidi, K Jbilou, L Reichel, and H Sadok. An extrapolated tsvd method for linear discrete ill-posed problems with kronecker structure. *Linear algebra and its applications*, 434(7):1677–1688, 2011.

[18] Abderrahman Bouhamidi, Mohammed Bellalij, Rentsen Enkhbat, Khalid Jbilou, and Marcos Raydan. Conditional gradient method for double-convex fractional programming matrix problems. *Journal of Optimization Theory and Applications*, 176(1):163–177, 2018.

[19] Abderrahman Bouhamidi, Rentsen Enkhbat, and Khalide Jbilou. Conditional gradient Tikhonov method for a convex optimization problem in image restoration. *Journal of Computational and Applied Mathematics*, 255:580–592, 2014.

[20] Abderrahman Bouhamidi and Khalide Jbilou. A kronecker approximation with a convex constrained optimization method for blind image restoration. *Optimization Letters*, 6(7):1251–1264, 2012.

[21] Alan C Bovik. *Handbook of image and video processing*. Academic press, 2010.

[22] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[23] Michael Brazell, Na Li, Carmeliza Navasca, and Christino Tamon. Solving multilinear systems via tensor inversion. *SIAM Journal on Matrix Analysis and Applications*, 34(2):542–570, 2013.

[24] Claude Brezinski and M Redivo Zaglia. *Extrapolation methods: theory and practice*. Elsevier, 2013.

[25] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science and Business Media, 2010.

[26] Daniela Calvetti and Lothar Reichel. Tikhonov regularization of large linear problems. *BIT Numerical Mathematics*, 43(2):263–283, 2003.

[27] Emmanuel J Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[28] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1):89–97, 2004.

[29] Raymond Chan, Tony Chan, and Andy Yip. Numerical methods and applications in total variation image restoration. *Handbook of mathematical methods in imaging*, 2015.

[30] Tony F Chan, Gene H Golub, and Pep Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM journal on scientific computing*, 20(6):1964–1977, 1999.

[31] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE transactions on Image Processing*, 7(3):370–375, 1998.

[32] Jean-Paul Chehab and Marcos Raydan. Geometrical inverse matrix approximation for least-squares problems and acceleration strategies. *Numerical Algorithms*:1–19, 2020.

[33] Chuan Chen, Zhe-Bin Wu, Zi-Tai Chen, Zi-Bin Zheng, and Xiong-Jun Zhang. Auto-weighted robust low-rank tensor completion via tensor-train. *Information Sciences*, 567:100–115, 2021.

[34] Edwin KP Chong and Stanislaw H Zak. *An introduction to optimization*. John Wiley & Sons, 2004.

[35] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[36] David Colton, Michele Piana, and Roland Potthast. A simple method using morozov's discrepancy principle for solving inverse scattering problems. *Inverse Problems*, 13(6):1477, 1997.

[37] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[38] Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[39] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image restoration by sparse 3d transform-domain collaborative filtering. In *Image Processing: Algorithms and Systems VI*, volume 6812, page 681207. International Society for Optics and Photonics, 2008.

[40] Philip J Davis. *Circulant matrices*. American Mathematical Soc., 2013.

[41] Jean-Pierre Dedieu. *Points fixes, zéros et la méthode de Newton*. Springer, 2006.

[42] Jiu Ding and LJ Huang. On the continuity of generalized inverses of linear operators in hilbert spaces. *Linear algebra and its applications*, 262:229–242, 1997.

[43] I Ekeland and R Temam. Convex analysis and variational problems (studies in mathematics and its applications, vol. 1 north-holland, amsterdam, 1976.

[44] Alaa El Ichi, Khalide Jbilou, and Rachid Sadaka. Tensor global extrapolation methods using the n-mode and the Einstein products. *Mathematics*, 8(8):1298, 2020.

[45] Heinz W Engl and Wilhelm Grever. Using the l–curve for determining optimal regularization parameters. *Numerische Mathematik*, 69(1):25–31, 1994.

[46] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[47] DA Fish, AM Brinicombe, ER Pike, and JG Walker. Blind deconvolution by means of the richardson–lucy algorithm. *JOSA A*, 12(1):58–65, 1995.

[48] Jens Flemming. *Variational Source Conditions, Quadratic Inverse Problems, Sparsity Promoting Regularization: New Results in Modern Theory of Inverse Problems and an Application in Laser Optics*. Springer, 2018.

[49] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[50] Roland W Freund and Manish Malhotra. A block qmr algorithm for non-hermitian linear systems with multiple right-hand sides. *Linear Algebra and its Applications*, 254(1-3):119–157, 1997.

[51] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & mathematics with applications*, 2(1):17–40, 1976.

[52] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[53] Claude Gasquet and Patrick Witomski. *Analyse de Fourier et applications: filtrage, calcul numérique, ondelettes*. Elsevier Masson, 1990.

[54] Pascal Getreuer. Rudin-osher-fatemi total variation denoising using split bregman. *Image Processing On Line*, 2:74–95, 2012.

[55] Pascal Getreuer. Total variation inpainting using split Bregman. *Image Processing On Line*, 2:147–157, 2012.

[56] Enrico Giusti. Minimal surfaces and functions of bounded variation. *Monogr. Math.*, 80, 1984.

[57] Roland Glowinski. *Numerical methods for nonlinear variational problems*. Tata Institute of Fundamental Research, 1980.

[58] Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.

[59] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, 21(1):185–194, 1999.

[60] Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[61] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2012.

[62] Gerardo González, Ville Kolehmainen, and Aku Seppänen. Isotropic and anisotropic total variation regularization in electrical impedance tomography. *Computers & Mathematics with Applications*, 74(3):564–576, 2017.

[63] Markus Grasmair, Markus Haltmeier, and Otmar Scherzer. Sparse regularization with lq penalty term. *Inverse Problems*, 24(5):055020, 2008.

[64] M El Guide, A El Ichi, FP Beik, and Khalide Jbilou. Tensor gmres and golub-kahan bidiagonalization methods via the einstein product with applications to image and video processing. *arXiv preprint arXiv:2005.07458*, 2020.

[65] M El Guide, A El Ichi, K Jbilou, and R Sadaka. Tensor krylov subspace methods via the t-product for color image processing. *arXiv preprint arXiv:2006.07133*, 2020.

[66] Mustapha Hached, Khalide Jbilou, Christos Koukouvinos, and Marilena Mitrouli. A multidimensional principal component analysis via the c-product golub–kahan–svd for classification and face recognition. *Mathematics*, 9(11):1249, 2021.

[67] Ali Haddad. Stability in a class of variational methods. *Applied and Computational Harmonic Analysis*, 23(1):57–73, 2007.

[68] Martin Hanke. Limitations of the l-curve method in ill-posed problems. *BIT Numerical Mathematics*, 36(2):287–301, 1996.

[69] Per Christian Hansen. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM review*, 34(4):561–580, 1992.

[70] Per Christian Hansen. *Discrete inverse problems: insight and algorithms*. SIAM, 2010.

[71] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998.

[72] Per Christian Hansen. The truncatedsvd as a method for regularization. *BIT Numerical Mathematics*, 27(4):534–553, 1987.

[73] Per Christian Hansen, Jakob Jørgensen, and William RB Lionheart. *Computed Tomography: Algorithms, Insight, and Just Enough Theory*. SIAM, 2021.

[74] Per Christian Hansen, James G Nagy, and Dianne P O'leary. *Deblurring images: matrices, spectra, and filtering*, volume 3. Siam, 2006.

[75] BS He, Hai Yang, and SL Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and applications*, 106(2):337–356, 2000.

[76] Min-Cheol Hong. Iterative adaptive hybrid image restoration for fast convergence. *Optical Engineering*, 49(8):080503, 2010.

[77] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[78] Baohua Huang and Changfeng Ma. Global least squares methods based on tensor form to solve a class of generalized sylvester tensor equations. *Applied Mathematics and Computation*, 369:124892, 2020.

[79] Kazufumi Ito and Bangti Jin. *Inverse problems: Tikhonov theory and algorithms*, volume 22. World Scientific, 2014.

[80] Kazufumi Ito and Karl Kunisch. *Lagrange multiplier approach to variational problems and applications*. SIAM, 2008.

[81] VK Ivanov. Conditions for well-posedness in the hadamard sense in spaces of generalized functions. *Siberian Mathematical Journal*, 28(6):906–911, 1987.

[82] Anil K Jain. *Fundamentals of digital image processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[83] Sindhu Jain and Mr Sudhir Goswami. A comparative study of various image restoration techniques with different types of blur. *International Journal of research in computer applications and robotics (IJRCAR)*, 3(11):54–60, 2015.

[84] K Jbilou and A Messaoudi. Block extrapolation methods with applications. *Applied Numerical Mathematics*, 106:154–164, 2016.

[85] K Jbilou and H Sadok. Vector extrapolation methods. applications and numerical comparison. *Journal of computational and applied mathematics*, 122(1-2):149–165, 2000.

[86] K Jbilou, H Sadok, and A Tinzefte. Oblique projection methods for linear systems with multiple right-hand sides. *Electronic Transactions on Numerical Analysis*, 20:119–138, 2005.

[87] Khalide Jbilou, Lothar Reichel, and Hassane Sadok. Vector extrapolation enhanced TSVD for linear discrete ill-posed problems. *Numerical Algorithms*, 51(2):195–208, 2009.

[88] Julie Kamm and James G Nagy. Kronecker product and svd approximations in image restoration. *Linear Algebra and its Applications*, 284(1-3):177–192, 1998.

[89] Julie Kamm and James G Nagy. Optimal kronecker product approximation of block toeplitz matrices. *SIAM Journal on Matrix Analysis and Applications*, 22(1):155–172, 2000.

[90] William J Kammerer and M Zuhair Nashed. Iterative methods for best approximate solutions of linear integral equations of the first and second kinds. *Journal of Mathematical Analysis and Applications*, 40(3):547–573, 1972.

[91] Saeed Karimi and Maryam Dehghan. Global least squares method based on tensor form to solve linear systems in kronecker format. *Transactions of the institute of measurement and control*, 40(7):2378–2386, 2018.

[92] Nikolaos P Galatsanos SAAAS Aggelos Katsaggelos. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their... *IEEE Transactions on image processing*, 1(3), 1992.

[93] Daniil Kazantsev, Jakob S Jørgensen, Martin S Andersen, William RB Lionheart, Peter D Lee, and Philip J Withers. Joint image reconstruction method with correlative multi-channel prior for x-ray spectral computed tomography. *Inverse Problems*, 34(6):064001, 2018.

[94] JT Kent and M Mohammadzadeh. Global optimization of the generalized cross-validation criterion. *Statistics and Computing*, 10(3):231–236, 2000.

[95] Eric Kernfeld, Misha Kilmer, and Shuchin Aeron. Tensor–tensor products with invertible linear transforms. *Linear Algebra and its Applications*, 485:545–570, 2015.

[96] Misha E Kilmer and Carla D Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[97] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[98] Tamara Gibson Kolda. Multilinear operators for higher-order decompositions. Technical report, Citeseer, 2006.

[99] Alexey Nikolaevich Krylov. On the numerical solution of the equation, which in technical matters determines the frequencies of small oscillations of material systems. *Proceedings of the Russian Academy of Sciences. Series mathematical*, (4):491–539, 1931.

[100] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.

[101] Chang Woo Lee, Keechul Jung, and Hang Joon Kim. Automatic text detection and removal in video sequences. *Pattern Recognition Letters*, 24(15):2607–2623, 2003.

[102] Namgil Lee and Andrzej Cichocki. Fundamental tensor operations for large-scale data analysis in tensor train formats. *arXiv preprint arXiv:1405.7786*, 2014.

[103] Namgil Lee and Andrzej Cichocki. Fundamental tensor operations for large-scale data analysis using tensor network formats. *Multidimensional Systems and Signal Processing*, 29(3):921–960, 2018.

[104] Adrien Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes; par AM Legendre...* chez Firmin Didot, libraire pour lew mathematiques, la marine, l . . ., 1806.

[105] Aleksandr Sergeevich Leonov. On the h-property of functionals in sobolev spaces. *Mathematical Notes*, 77(3):348–363, 2005.

[106] Chengbo Li. *An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing*. PhD thesis, Rice University, 2010.

[107] Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in neural information processing systems*, pages 379–387, 2015.

[108] Xutao Li, Yunming Ye, and Xiaofei Xu. Low-rank tensor completion with total variation for visual data inpainting. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2210–2216, 2017.

[109] Youzuo Lin and Brendt Wohlberg. Application of the upre method to optimal parameter selection for large scale regularization problems. In *2008 IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 89–92. IEEE, 2008.

[110] Youzuo Lin, Brendt Wohlberg, and Hongbin Guo. Upre method for total variation parameter selection. *Signal Processing*, 90(8):2546–2551, 2010.

[111] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2012.

[112] Williams López and Marcos Raydan. An acceleration scheme for Dykstra's algorithm. *Computational Optimization and Applications*, 63(1):29–44, 2016.

[113] Cécile Louchet. *Modèles variationnels et bayésiens pour le débruitage d'images: de la variation totale vers les moyennes non-locales*. PhD thesis, Université René Descartes-Paris V, 2008.

[114] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):925–938, 2019.

[115] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Exact low tubal rank tensor recovery from gaussian measurements. *arXiv preprint arXiv:1806.02511*, 2018.

[116] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[117] Uday Modha and Preeti Dave. Image inpainting-automatic detection and removal of text from images. *International Journal of Engineering Research and Applications (IJERA), ISSN*:2248–9622, 2014.

[118] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

[119] Vladimir Alekseevich Morozov. On the solution of functional equations by the method of regularization. In *Doklady Akademii Nauk*, volume 167 of number 3, pages 510–512. Russian Academy of Sciences, 1966.

[120] Jennifer L Mueller and Samuli Siltanen. *Linear and nonlinear inverse problems with practical applications*. SIAM, 2012.

[121] MZ Nashed. A new approach to classification and regularization of ill-posed operator equations. In *Inverse and ill-posed problems*, pages 53–75. Elsevier, 1987.

[122] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[123] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[124] Christopher C Paige and Michael A Saunders. Lsqr: an algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):43–71, 1982.

[125] Christopher C Paige and Michael A Saunders. Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis*, 12(4):617–629, 1975.

[126] Christopher C Paige and Michael A Saunders. Towards a generalized singular value decomposition. *SIAM Journal on Numerical Analysis*, 18(3):398–405, 1981.

[127] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[128] Michael JD Powell. A method for nonlinear constraints in minimization problems. *Optimization*:283–298, 1969.

[129] Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: consistency and scalable methods. *Advances in neural information processing systems*, 28, 2015.

[130] Dongwei Ren, Hongzhi Zhang, David Zhang, and Wangmeng Zuo. Fast total-variation based image restoration based on derivative alternated direction optimization methods. *Neurocomputing*, 170:201–212, 2015.

[131] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

[132] Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, 2015. ISBN: 9781400873173. DOI: doi:10.1515/9781400873173. URL: https://doi.org/10.1515/9781400873173.

[133] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[134] Fredrik Rusek, Daniel Persson, Buon Kiong Lau, Erik G Larsson, Thomas L Marzetta, Ove Edfors, and Fredrik Tufvesson. Scaling up mimo: opportunities and challenges with very large arrays. *IEEE signal processing magazine*, 30(1):40–60, 2012.

[135] Youcef Saad and Martin H Schultz. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[136] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[137] Oguz Semerci, Ning Hao, Misha E Kilmer, and Eric L Miller. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. *IEEE Transactions on Image Processing*, 23(4):1678–1693, 2014.

[138] Avram Sidi. Efficient implementation of minimal polynomial and reduced rank extrapolation methods. *Journal of computational and applied mathematics*, 36(3):305–337, 1991.

[139] Nicholas D Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):229–239, 2000.

[140] G Sridevi and S Srinivas Kumar. Image inpainting based on fractional-order nonlinear diffusion for image reconstruction. *Circuits, Systems, and Signal Processing*, 38(8):3802–3817, 2019.

[141] Xue-Cheng Tai and Chunlin Wu. Augmented lagrangian method, dual methods and split bregman iteration for rof model. In *International conference on scale space and variational methods in computer vision*, pages 502–513. Springer, 2009.

[142] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[143] Jan Van Leeuwen and Jan Leeuwen. *Handbook of theoretical computer science: Algorithms and complexity*, volume 1. Elsevier, 1990.

[144] Charles F Van Loan and Nikos Pitsianis. Approximation with Kronecker products. In *Linear algebra for large scale and real-time applications*, pages 293–314. Springer, 1993.

[145] Guorong Wang, Yimin Wei, and Sanzheng Qiao. *Moore-penrose inverse of linear operators*. In *Generalized Inverses: Theory and Computations*. Springer Singapore, Singapore, 2018, pages 317–338. ISBN: 978-981-13-0146-9. DOI: 10.1007/978-981-13-0146-9_11. URL: https://doi.org/10.1007/978-981-13-0146-9_11.

[146] Yanfei Wang, Anatoly G Yagola, Changchun Yang, et al. *Optimization and regularization for computational inverse problems and applications*. Springer, 2011.

[147] Brendt Wohlberg. Admm penalty parameter selection by residual balancing. *arXiv preprint arXiv:1704.06209*, 2017.

[148] Chunlin Wu, Juyong Zhang, and Xue-Cheng Tai. Augmented lagrangian method for total variation restoration with non-quadratic fidelity. *Inverse Problems & Imaging*, 5(1):237–261, 2011.

[149] Peiliang Xu. Truncated svd methods for discrete linear ill-posed problems. *Geophysical Journal International*, 135(2):505–514, 1998.

[150] Tatsuya Yokota, Qibin Zhao, and Andrzej Cichocki. Smooth parafac decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20):5423–5436, 2016.

[151] Tjalling J Ypma. Historical development of the newton–raphson method. *SIAM review*, 37(4):531–551, 1995.

[152] Jian Zhang, Debin Zhao, Ruiqin Xiong, Siwei Ma, and Wen Gao. Image restoration using joint statistical modeling in a space-transform domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6):915–928, 2014.

[153] Zemin Zhang and Shuchin Aeron. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2016.

[154] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3842–3849, 2014.

[155] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1751–1763, 2015.

[156] Pan Zhou, Canyi Lu, Zhouchen Lin, and Chao Zhang. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3):1152–1163, 2017.

[157] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, 34, 2008.

Regularization of tensorial inverse problems via convex optimization
Applications in image and video processing

Abstract

Solving inverse problems in a multidimensional setting has become an active topic on which many researchers in linear algebra are working. On the one hand, the construction of a higher dimensional model can be achieved using tensor algebra by adopting the mechanisms developed recently in this field. On the other hand, the solution to such problems is usually based on the regularization techniques that remedy the ill-conditioning that can be exhibited in almost all inverse problems.
The present thesis aims to bring together the modeling of inverse problems in a higher dimension and the generalization of some variational regularization methods in tensorial form. Recently, the variational regularization methods are known as well-established methods for solving inverse problems. For example, Tikhonov and total variation regularizers are among the well-known approaches that we will generalize and develop in the tensor form. Convex optimization approaches will play an essential role in the resolution of the constrained regularization problems that we have proposed. As well as a set of mechanisms, such as projection methods and extrapolation techniques, which have contributed to enhancing the performance of the developed algorithms. Numerical applications in image and video processing are given to illustrate the effectiveness of the proposed approaches compared with some state-of-art methods.

Keywords: tensorial algebra, convex optimization, regularization methods, Tickonov, total variation, projection methods, extrapolation techniques, image and video processing.

Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville (LMPA) – 50, rue Ferdinand Buisson – CS 80699 – 62228 Calais Cedex – France
Laboratoire de Mathématiques Appliquées et Informatique (LAMAI) – UCAM, FSTG, – BP 549 Marrakech – Maroc

Régularisation des problèmes inverses tensoriels par l'optimisation convexe
Applications au traitement des images et des vidéos

Résumé

La résolution des problèmes inverses dans un cadre multidimensionnel est devenue un sujet actif sur lequel travaillent de nombreux chercheurs en algèbre linéaire. D'une part, la construction d'un modèle en dimension supérieure peut être réalisée en utilisant l'algèbre tensorielle en adoptant les mécanismes développés récemment dans ce domaine. D'autre part, la résolution de tels problèmes est généralement basée sur l'utilisation de techniques de régularisation qui remédient au mauvais conditionnement que l'on peut trouver dans presque tous les problèmes inverses.
La présente thèse vise à réunir la modélisation des problèmes inverses en dimension supérieure et la généralisation de certaines méthodes de régularisation variationnelle sous forme tensorielle. Les méthodes de régularisation variationnelle sont connues comme des méthodes bien établies pour résoudre les problèmes inverses. Par exemple, les régularisateurs de Tikhonov et de la variation totale font partie des approches bien connues que nous généraliserons et développerons sous forme tensorielle. Les approches d'optimisation convexe joueront un rôle essentiel dans la résolution des problèmes de régularisation sous contraintes que nous avons proposés. Ainsi qu'un ensemble de mécanismes, tels que les méthodes de projection et les techniques d'extrapolation, qui ont contribué à améliorer les performances des approches développées. Des applications numériques dans le traitement des images et des vidéos sont données pour illustrer l'efficacité des approches proposées par rapport à certaines méthodes de l'état de l'art.

Mots clés : algèbre tensorielle, optimisation convexe, méthodes de régularisation, Tickonov, variation totale, méthodes de projection, techniques d'extrapolation, traitement des images et des vidéos.