Krylov Subspace Methods for Inverse Problems with Application to Image Restoration

Mohamed El Guide

Krylov Subspace Methods for Inverse Problems with Application to Image Restoration

Mohamed El Guide

A DISSERTATION

in

Mathematics

Presented to the Faculties of the University of Cadi Ayyad in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

26/12/2017

Supervisors of Dissertation

_____

Abdeslem Hafid Bentib, Professor of Mathematics
Khalide Jbilou, Professor of Mathematics

Graduate Group Chairperson

_____

Hassane Sadok, Université du Littoral-Côte-d'Opale

Dissertation Committee :
Nour Eddine Alaa , Faculté des Sciences et Techniques de Marrakech
Abdeslem Hafid Bentbib, Faculté des Sciences et Techniques de Marrakech
Mohamed El Alaoui Talibi, Faculté des Sciences Semlalia de Marrakech
Said El Hajji, Faculté des Sciences de Rabat
Khalide Jbilou, Université du Littoral-Côte-d'Opale
Abderrahim Messaoudi, ENS Rabat

Acknowledgments

First of all, I would like to thank the people who believed in me and who allowed me to arrive till the end of this thesis. A special thanks goes to my advisors Abdeslem Hafid Bentib and Khalide Jbilou, for their guidance, support, and encouragement during the period of my preparation to this thesis. I would also like to express my sincere gratitude to Professor Lothar Reichel. He contributed with an immense amount of valuable inputs to improve this dissertation. He always opens the door to reflect his willingness to give advice, discuss research, and share his passion on numerical linear algebra at any time. I would like to express my appreciation to all of my jury members : Pr. Nour Eddine Alaa, Pr. Mohamed El Alaoui Talibi, Pr. Said El Hajji, Pr. Abderrahim Messaoudi and Pr. Hassane Sadok. Special mention goes to Pr. Nour Eddine Alaa, Pr. Abderrahim Messaoudi and Pr. Hassan Sadok who supported me and allowed me to motivated raising, light heart and quiet mind since the beginning of my thesis. I extend my gratitude to all my friends and to all those who have helped in carrying out this work. Finally, the simplest words being the strongest, I address all my affection to my family, and in particular to my mother who made me understand that life is not not only problems that could be solved through mathematical formulas and algorithms. Despite the distance from my family for many years, their intelligence, their confidence, their tenderness, their love carry me and guide

me all days. Thank you for making me what I am today. I want to thank Abdelghani, Nezha, Kamal, Karim and Zagour who made me redefined the meaning of friendship.

ABSTRACT

Mohamed El Guide

Abdeslem Hafid Bentbib, Khalide Jbilou

Image restoration often requires the solution of large linear systems of equations with a very ill-conditioned, possibly singular, matrix and an error-contaminated right-hand side. The latter represents the available blur and noise-contaminated image, while the matrix models the blurring. Computation of a meaningful restoration of the available image requires the use of a regularization method. We consider the situation when the blurring matrix has a Kronecker product structure and an estimate of the norm of the desired image is available, and illustrate that efficient restoration of the available image can be achieved by Tikhonov regularization based on the global Lanczos method, and by using the connection of the latter to Gauss-type quadrature rules. We also investigate the use of the global Golub-Kahan bidiagonalization method to reduce the given large problem to a small one. The small problem is solved by employing Tikhonov regularization. A regularization parameter determines the amount of regularization. The connection between global Golub-Kahan bidiagonalization and Gauss-type quadrature rules is exploited to inexpensively compute bounds that are useful for determining the regularization parameter by the discrepancy principle. We will also present an efficient algorithm for solving the Tikhonov regularization problem of a linear system of equations with multiple right-hand sides contaminated by errors. The proposed algorithm is based on the symmetric block Lanczos algorithm, in connection with block Gauss quadrature rules. We will show how this connection is designed to inexpensively determine a value of the regularization pa-

rameter when a solution norm constraint is given. Next, we will present four algorithms for the solution of linear discrete ill-posed problems with several right-hand side vectors. These algorithms can be applied, for instance, to multi-channel image restoration when the image degradation model is described by a linear system of equations with multiple right-hand sides that are contaminated by errors. Two of the algorithms are block generalizations of the standard Golub-Kahan bidiagonalization method with the block size equal to the number of channels. One algorithm uses standard Golub-Kahan bidiagonalization without restarts for all right-hand sides. These schemes are compared to standard Golub-Kahan bidiagonalization applied to each right-hand side independently. Tikhonov regularization is used to avoid severe error propagation. Applications include the restoration of color images are given. We will finally give efficient algorithms to solve total variation (TV) regularization of images contaminated by blur and additive noise. The unconstrained structure of the problem suggests that one can solve a constrained optimization by transforming the original unconstrained minimization problem to an equivalent constrained minimization problem. An augmented Lagrangian method is used to handle the constraints, and an alternating direction method (ADM) is used to iteratively find solutions of the subproblems. The solution of these subproblems are belonging to subspaces generated by application of successive orthogonal projections onto a class generalized matrix Krylov subspaces of increasing dimension.

Table des matières

Table des figures

Liste des tableaux

Introduction

Le travail de cette thèse rentre dans le cadre générale de l'algèbre linéaire et l'analyse numérique. Le principale objectif est le développement de méthodes numériques matricielles pour la résolution des problèmes inverses de grande taille, ainsi que l'application de ces méthodes à des problèmes importants qui se produisent dans la science et l'industrie. Les méthodes numériques sont développées, analysées et spécialisées pour résoudre des problèmes discrets mal-posés de grande taille. Dans cette thèse, nous nous concentrons sur des problèmes en deux dimensions. Nous rencontrons de tels problèmes dans diverses applications et tout particulièrement dans la résolution des équations intégrales de Fredholm du premier type en deux dimensions avec un noyau séparable [13], ainsi que la restauration d'images floues et bruitées dans le cas où la fonction de l'étalement du point est séparable [62].

La restauration de l'image

La restauration d'image est une étape très importante dans le processus de traitement d'image. Cette dernière est nécessaire pour corriger les différents traitements que peut subir une image, car celle- ci, pendant sa capture peut être, dans la plus part des

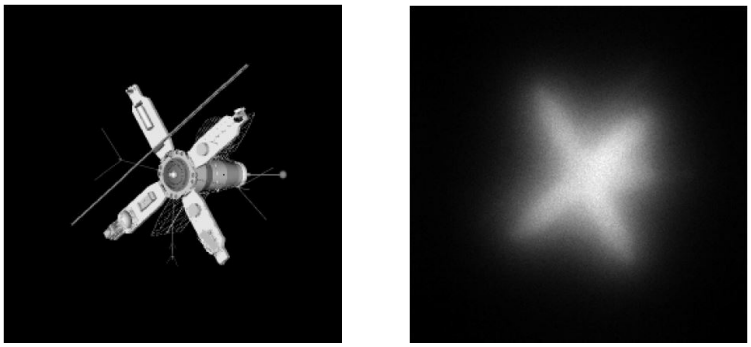
cas, sujette à des dégradations qui conduisent aux manques d'informations utiles qu'elle contenait et qui sont d'une grande importance.

Au cours des années 50, une grande importance a été attribuée aux techniques numériques de restauration d'images, et cela est dû essentiellement à la mauvaise qualité des images que les astronautes prélevaient. Cette qualité est influencée par plusieurs sources de dégradation qui peuvent être, soit l'appareil de prise de vue, soit l'opérateur, soit l'objectif, ou même les turbulences atmosphériques. Avec le développement technologique, plusieurs domaines vitaux (médical, militaire,...etc.) essaient d'en profiter de cette technologie, et principalement le domaine de l'imagerie, ce qui nécessite automatiquement des outils de plus en plus sophistiqués qui répondent aux besoins des opérateurs. Lorsqu'on utilise un appareil photo, nous voulons que l'image enregistrée soit une représentation exacte de la scène que nous voyons, mais chaque image est plus ou moins floue. Ainsi, la restauration de l'image est fondamental dans la prise de photos nettes .

La représentation la plus utilisée d'une image numérique est celle d'un tableau à deux dimensions composé d'un ensemble de lignes et de colonnes. Chaque cellule du tableau, appelée pixel, contient une valeur quantifiée. Cette valeur est une sémantique dépendant du type de signal qu'elle code (intensité lumineuse du point, distance à un point de référence, ou numéro de la région d'appartenance par exemple). Une petite image a environ 2562 à 65536 pixels tandis qu'une image de haute résolution souvent a 5 à 10 millions de pixels. Le flou s'apparait toujours dans l'enregistrement d'une image numérique, car il est inévitable que les informations de la scène débordent aux pixels voisins. La restauration de l'image est le processus d'enlever le flou et le bruit de l'image dégradée pour récupérer une approximation de l'image originale. Ce domaine de la technologie d'imagerie est de-

venu de plus en plus important dans de nombreuses applications scientifiques telles que l'astronomie [15, 7, 68], l'imagerie médicale [15, 60, 94], militaires [15, 83], la surveillance [15, 83], la microscopie [29, 8] et la technologie de communication vidéo [15, 8].

Par exemple, les scientifiques utilisent des télescopes pour obtenir les images des étoiles et des planètes lointaines. Toutefois, en raison de la déformation causée par l'atmosphère de la terre et les rayons de la lumière aléatoire provenant de diverses sources, les astronomes reçoivent des images floues, comme celle représentée dans la Figure 1.1. De même, les



(a) Image originale (b) Image floue

FIGURE 1 – Image satellite.

médecins et les technologues médicaux obtiennent des images de l'anatomie humaine et la physiologie à partir des machines radiologiques tels que X-ray, imagerie par résonance magnétique (IRM). Puisque le dispositif de la formation de l'image est situé à l'extérieur du patient, le corps sert de distorsions moyennes, avec le rayonnement aléatoire, peuvent corrompre les images. En outre, le mouvement du patient peut provoquer plus de flou dans l'image. Cependant, le bruit et le flou peut être filtré de ces images, ce qui les rend plus facile à déchiffrer par les médecins.

La surveillance est un autre domaine fortement influencé par les technologies de la res-

tauration de l'image. Par exemple, les responsables de l'application de la loi et la police scientifique utilisent des techniques de la restauration numérique pour récupérer des visages et des plaques d'immatriculation à partir des cassettes vidéo de sécurité de mauvaise qualité. Plus récemment, des techniques telles que le masque de phase cubique sont utilisés pour apporter tous les visages dans une foule, peu importe leur proximité de la caméra [29].

Dans chacune de ces applications, obtenir des images plus claires peut être accompli en utilisant des programmes informatiques pour effectuer des techniques d'amélioration de l'image. Ce processus de calcul peut être compliqué, nécessitant des algorithmes pour résoudre des dizaines de milliers, peut-être des millions d'équations mathématiques. Cependant, une reconstruction précise et efficace peut être extrêmement avantageuse.

Considérons le modèle linéaire de le restauration d'image donné par l'équation de Fredholm du premier type en deux dimensions suivante

$$\int\int_{\Omega} \kappa(x,y,s,t) f(s,t) ds dt = g(x,y), \qquad (x,y) \in \Omega', \tag{0.0.1}$$

où κ est le noyau, f est l'image originale et g est l'image floue et bruitée. La restauration en utilisant un système linéaire est l'hypothèse principale dans cette thèse, donc le processus de dégradation, c'est à dire, l'opération de passer de l'image nette à l'image floue, est linéaire. Dans les sciences physiques, cette hypothèse est faite parce que, dans de nombreux cas, le flou est en effet linéaire, ou au moins bien approché par un modèle linéaire. Une conséquence importante de cette hypothèse est que nous avons un grand nombre d'outils de l'algèbre linéaire et de calcul matriciel à notre disposition. La clé pour obtenir ce modèle linéaire est de réarranger les éléments des images $X \in \mathbb{R}^{m\times n}$ (discrétisation de f)

et $B \in \mathbb{R}^{m\times n}$ (discrétisation de g) dans des vecteurs colonnes en empilant les colonnes de celles-ci en deux vecteurs x et b , à la fois de longueur $N = mn$. Comme le flou est supposé une opération linéaire, il doit exister une grande matrice de flou $H \in \mathbb{R}^{N\times N}$ tel que x et b sont liés par le modèle linéaire

$$Hx = b. \tag{0.0.2}$$

L'image disponible, représentée par b, est supposé être contaminé par le flou et le bruit. Soit $e \in \mathbb{R}^{mn}$ le bruit, e est normalisé avec une moyenne et variance nulles. Alors b peut s'exprimer par $b = \widehat{b} + e$, où $\widehat{b}$ est le second membre du système linéaire $H\widehat{x} = \widehat{b}$ associé à l'image originale $\widehat{x}$. Généralement, la résolution du problème (0.0.2) nécessite un traitement des données de grande taille et alors l'allocation de larges plages de mémoire. Heureusement, dans de nombreuses applications, le noyau satisfait $\kappa(x, y; s, t) = \kappa(x - s, y - t)$, et on dit qu'il est spatialement invariant ou isotrope. Dans ce cas, l'équation intégrale (0.0.1) est une opération de convolution, et donc κ peut généralement être représenté par une structure de données compacte lorsque le stockage devient un problème. Les applications où le noyau est spatialement invariant et alors exploiter la structure de la matrice H peuvent être trouvées dans [5, 37, 80, 82]. La résolution du problème vectoriel (0.0.2) nécessite l'utilisation d'un schéma itératif [87, 57], la partie la plus intensive de ces méthodes est les multiplications matrice-vecteur avec H, ce qui est supposé être extrêmement numériquement couteux, donc un schéma de stockage efficace en exploitant la structure devrait être utilisé. Un cas particulier se produit lorsque κ est séparable, c'est-à-dire $\kappa(x-s, y-t) = \kappa_1(x-s)\kappa_2(y-t)$. Si H_1 et H_2 sont les discrétisations des opérateurs intégraux κ_1 et κ_2, respectivement, alors l'opérateur discret correspondant H du noyau

κ est défini par $H = H_1 \otimes H_2$, où $\otimes$ désigne le produit de Kronecker [55]. L'objectif de cette thèse est de développer des méthodes numériques efficaces exploitant la structure du noyau κ. À ce stade, nous supposerons que la matrice floue H est le produit de Kronecker de deux matrices. Notons qu'il est bien connu que la plupart des matrices de flou ont la structure de Kronecker ou peuvent être bien approximées par des matrices ayant cette structure ; voir, par exemple, Kamm et Nagy [71, 72] et Van Loan et Pitsianis [99]. En inversant la matrice H, une approximation de la solution exacte peut être calculée ; Cependant, ce n'est pas si simple à cause du mauvais conditionnement de la matrice H. Les méthodes de régularisation peuvent être utilisées pour calculer une approximation de x. L'une des méthodes de régularisation les plus connues est la régularisation de Tikhonov.

Régularisation de Tikhonov

Puisque la matrice de flou H est mal conditionnée, le problème de la restauration d'image sera extrêmement sensible aux perturbations dans le second membre. Afin de diminuer l'effet du bruit dans les données, nous remplaçons le modèle (0.0.2) par un autre mieux conditionné. L'une des méthodes de régularisation les plus populaires est celle de Tikhonov [31, 49]. La méthode remplace le problème (0.0.2) par celui des moindres carrés suivant

$$\min_x \big(\|Hx - b\|_2^2 + \mu \|x\|_2^2 \big), \tag{0.0.3}$$

où $\mu > 0$ est le paramètre de régularisation. Les équations normales associées à (0.0.3) sont données par

$$(H^T H + \mu I)x = H^T b, \tag{0.0.4}$$

où I est la matrice d'identité de l'ordre approprié et l'exposant ^{T} indique la transposition. Il s'ensuit que (0.0.3) admet la solution unique

$$x_\mu := (H^T H + \mu I)^{-1} H^T b \tag{0.0.5}$$

pour tout $\mu > 0$. Dans [46, 47, 25], Gene Golub et ses collaborateurs ont développé des méthodes élégantes et efficaces basées sur la relation entre l'algorithme de Lanczos, les polynômes orthogonaux, les quadratures de Gauss et certaines fonctions matricielles pour calculer les paramètres de régularisation et approximer la solution du problème de régularisation de Tikhonov. Cependant, cette technique ne s'applique pas dans certains cas puisque la mise en oeuvre du problème de restauration d'image nécessite typiquement le traitement des données de très grande taille, donc cette approche peut être échouée en terme de stockage mémoire et temps d'exécution. Heureusement, ce coût peut être considérablement réduit en exploitant la structure de Kronecker de la matrice H. Au Chapitre 3, nous supposerons qu'une estimation de la solution exacte est donnée, et nous utiliserons cette estimation pour déterminer μ. Cette hypothèse présente l'avantage que, dans certaines applications, les propriétés physiques du problème déterminent une valeur optimale pour la contrainte de la norme. C'est le cas, par exemple, de la restauration d'image où la contrainte de la norme représente l'énergie de l'image cible. Cette approche pour sélectionner μ est préconisé par Rojas et Sorensen [91].

Puisque de nombreuses matrices floues peuvent être représentées, ou être bien approximées, par un produit de Kronecker ; voir, par exemple, Kamm et Nagy [71, 72] et Van Loan et Pitsiani [99]. Nous supposerons que H a une structure de produit de Kronecker et nous présentons un algorithme pour approximer la solution exacte en exploitant

cette structure. Nous montrerons comment la structure de produit Kronecker nous a permet d'utiliser la méthode de Lanczos globale. Cette méthode remplace l'évaluation des produits matrice-vecteur de l'algorithme de Lanczos standard par l'évaluation des produits matrice-matrice, qui peuvent être exécutés efficacement sur de nombreux ordinateurs modernes. La méthode de Lanczos globale est décrite dans [67]. Nous allons déterminer une valeur appropriée du paramètre de régularisation μ en estimant d'abord une certaine intégrale de Riemann-Stieltjes par des règles de quadrature de type Gauss. Ces règles peuvent être évaluées à l'aide de l'algorithme de Lanczos globale. Cette idée est une extension de la méthode de Lanczos standard utilisée pour déterminer μ [25].

Au Chapitre 4, nous utiliserons le *discrepancy principle* pour choisir le paramètre de régularisation μ. Cette méthode nécessite qu'une approximation ε de $\|e\|_2$ soit disponible et prescrit que $\mu > 0$ soit déterminé de sorte que $\|b - Hx_\mu\|_2 = \eta\varepsilon$ pour un choix de la constante $\eta \geq 1$ de la part de l'utilisateur, cette constante est indépendante de ε ; voir [31, 58, 90] pour des discussions sur cette méthode de choix des paramètres. Nous allons déterminer une valeur $\mu > 0$ telle que

$$\varepsilon \leq \|b - Hx_\mu\|_2 \leq \eta\varepsilon, \tag{0.0.6}$$

où la constante $\eta > 1$ est indépendante de ε. Le calcul d'une valeur μ telle que la solution associée x_μ satisfait (0.0.6) nécessite généralement l'utilisation d'une méthode de résolution d'une équation non linéaire, typiquement $\|b - Hx_\mu\|_2$ doit être évaluée pour plusieurs valeurs de μ. Cela peut être numériquement couteux quand la matrice H est de grande taille. Une méthode de résolution basée sur la réduction de H à une matrice bidiagonale de petite taille à l 'aide de la bidiagonisation de Golub-Kahan (Golub-Kahan

bidiagonalization ou GKB en anglais) est discutée dans [26]. Cette méthode utilise la relation entre les quadratures de type Gauss et GKB pour déterminer une approximation de x_μ qui satisfait (0.0.6). C'est alors notre objectif de décrire une méthode analogue pour la situation où H est le produit Kronecker de deux matrices. Notre approche est de remplacer la méthode GKB par la méthode de la bidiagonalisation de Golub - Kahan globale (Global Golub-Kahan bidiagonalization ou GGKB en anglais) décrite par Toutounian et Karimi [95]. L'évaluation des produits matrice-vecteur de grande taille dans la méthode GKB est remplacée par l'évaluation des produits matrice-matrice de petites taille par GGKB. Nous allons exploiter la relation entre les règles de quadrature de type Gauss et la méthode GGKB pour déterminer une valeur μ et une approximation associée du vecteur x_μ qui satisfait (0.0.6).

Au Chapitre 5, nous proposons une nouvelle méthode pour déterminer une valeur appropriée du paramètre de régularisation et une solution approximative associée, lors de la résolution d'un système linéaire d'équations mal conditionnés à plusieurs second membres contaminés par des erreurs. La méthode proposée est basée sur les algorithmes de Lanczos symétrique par bloc, en relation avec les règles de quadrature de Gauss par bloc pour approximer de façon qui n'est pas numériquement coûteuse une fonction matricielle de la forme $W^T f(H)W$, où $W \in \mathbb{R}^{n\times k}$, $k \ll n$, et $H \in \mathbb{R}^{n\times n}$ est une matrice symétrique, apparaissant lors de l'application de la régularisation de Tikhonov.

Le Chapitre 6 traite l'utilisation des méthodes itératives basées sur la bidiagonalisation standard ou par bloc de type Golub-Kahan, combinées avec la régularisation de Tikhonov pour la restauration d'une image de plusieurs canaux à partir d'une version floue et bruitée. Les applications incluent la restauration d'images couleurs dont la représentation

RVB (rouge, vert et bleu) utilise trois canaux ; voir [44, 62]. Les méthodes décrites peuvent également être appliquées à la solution des équations intégrales de Fredholm du premier type en deux ou plusieurs dimensions et à la restauration des images hyper-spectrale. Dans ce chapitre, nous nous concentrons sur la restauration d'images de k canaux qui ont été contaminées par un flou et bruit, et formuler cette tâche de restauration comme un système linéaire d'équations à k second membres, où chaque bande spectrale correspond à un canal. Pour simplifier les notations nous assumons que l'image est représentée par un tableau de $n \times n$ pixels dans chacun des k canaux où $1 \leq k \ll n^2$. Soit $b^{(i)} \in \mathbb{R}^{n^2}$ le vecteur qui représentent le flou et le bruit dans l'image contaminée i (le canal i), soit $e^{(i)} \in \mathbb{R}^{n^2}$ le vecteur décrivant le bruit dans ce canal, et soit $\widehat{x}^{(i)} \in \mathbb{R}^{n^2}$ le vecteur qui indique l'image inconnue i (canal i) sans flou et bruit. Les quantités correspondantes pour tous ces k canaux $b, \widehat{x}, e \in \mathbb{R}^{n^2 k}$ sont obtenus en empilant les vecteurs $b^{(i)}, \widehat{x}^{(i)}, e^{(i)}$ de chaque canal. Par exemple, $b = [(b^{(1)})^T, \ldots, (b^{(k)})^T]^T$.

La régularisation par la variation totale

La régularisation par la variation totale (Total variation ou TV en anglais) est l'une des stratégies les plus populaires et les plus efficaces pour la restauration d'images. Elle est bien connue pour préserver les bords et les discontinuités tout en créant des zones lisses. Rudin, Osher et Fatemi ont introduit TV dans [88]. Une discussion détaillée sur TV a été présenté dans [20, 23] par Chambolle et al. En raison de la non-differentiabilité et de la non-linéarité des modèles de régularisation par TV, les méthodes d'optimisation non linéaires sont nécessaires et elles sont donc plus exigeantes en termes de calcul que la simple

résolution des problèmes linéaires de régularisation de Tikhonov. Un effort massif a été fait en proposant des algorithmes efficaces capables de traiter les propriétés non linéaires des modèles de régularisation par TV ; voir par exemple [101, 19, 50, 54]. Cependant, ces algorithmes sont toujours soit beaucoup plus lents, soit moins robustes par rapport aux algorithmes conçus pour la minimisation des problèmes de régularisation de Tikhonov. Les algorithmes proposés dans le chapitre 7 de cette thèse ont réussi à surmonter cette difficulté et ont conduit à de nouveaux solveurs pour la minimisation par TV.

Chapitre 1

Preliminaries

This chapter introduces basic notions and operations in numerical linear algebra that will be used in the analysis of inverse problems.

1.1 Definitions and theorems

Definition 1.1.1. (Eigenvalues and eigenvectors). Given a square matrix $A \in \mathbb{R}^{n\times n}$, an eigenvector of A is a vector $v \neq 0 \in \mathbb{R}^n$ such that $Av = \lambda v$, $\lambda \in \mathbb{R}$. The scalar λ is called the eigenvalue corresponding to the eigenvector v.

Definition 1.1.2. (Normal matrix). A square matrix A with real coefficients is a normal matrix if it commutes with its transpose matrix A^T, that is to say if $AA^T = A^TA$.

Definition 1.1.3. (Orthogonal matrix). A real orthogonal matrix is a square matrix A with real entries whose columns and rows are orthogonal unit vectors i.e.

$$A^TA = AA^T = I, \tag{1.1.1}$$

where I is the identity matrix.

This leads to the equivalent characterization : a matrix A is orthogonal if its transpose is equal to its inverse :

$$A^T = A^{-1}. \tag{1.1.2}$$

Theorem 1.1.4. *(Spectral decomposition). Let $A \in \mathbb{R}^{n\times n}$ be a real square matrix. Then there exist an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$ such that*

$$A = Q\Lambda Q^T \tag{1.1.3}$$

where Λ contains the eigenvalues of A and the columns of Q are the corresponding (properly normalized) eigenvectors. The identity (1.1.3) is called the Spectral Decomposition or Eigendecomposition of the matrix A.

Definition 1.1.5. (Singular values and singular vectors). Given a matrix $A \in \mathbb{R}^{m\times n}$, a scalar $\sigma \geq 0$ is a singular value for A if and only if there exist unit-length vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$ such that

$$Av = \sigma u, \quad A^T u = \sigma v. \tag{1.1.4}$$

The vectors u and v are called left-singular and right-singular vectors for σ, respectively.

Theorem 1.1.6. *(Singular value decomposition). Let $A \in \mathbb{R}^{m\times n}$ be a matrix. Then there exist an orthogonal matrix $U \in \mathbb{R}^{m\times m}$, a diagonal matrix $\Sigma \in \mathbb{R}^{m\times n}$ and a unitary matrix $V \in \mathbb{R}^{n\times n}$ such that*

$$A = U\Sigma V^T \tag{1.1.5}$$

where Σ contains the singular values of A and the columns of U and V are the corresponding (properly normalized) left and right singular vectors, respectively. The identity (1.1.5) is called the Singular Value Decomposition (SVD) of the matrix A.

Theorem 1.1.7. *(QR decomposition). Let A be a square matrix. Then A may be decomposed as*

$$A = QR, \tag{1.1.6}$$

where Q is an orthogonal matrix (its columns are orthogonal unit vectors meaning $Q^TQ = I$) and R is an upper triangular matrix (also called right triangular matrix). If A is invertible, then the factorization is unique if we require the diagonal elements of R to be positive.

1.2 Kronecker product

Definition 1.2.1. (Kronecker product). If H_2 is an $m \times n$ matrix and $H_1 = [h_{i,j}^{(1)}]$ is a $p \times q$ matrix, then the Kronecker product $H_1 \times H_2$ is the $mp \times nq$ block matrix :

$$H = H_1 \otimes H_2 = \begin{bmatrix} h_{1,1}^{(1)}H_2 & h_{1,2}^{(1)}H_2 & \cdots & h_{1,q}^{(1)}H_2 \\ h_{2,1}^{(1)}H_2 & h_{2,2}^{(1)}H_2 & \cdots & h_{2,q}^{(1)}H_2 \\ \vdots & \vdots & & \vdots \\ h_{p,1}^{(1)}H_2 & h_{p,2}^{(1)}H_2 & \cdots & h_{p,q}^{(1)}H_2 \end{bmatrix}. \tag{1.2.1}$$

Definition 1.2.2. (vec and mat operators) The vec operator transforms a matrix $A = [a_{i,j}] \in \mathbb{R}^{m\times n}$ to a vector $a \in \mathbb{R}^{mn}$ by stacking the columns of A from left to right, i.e,

$$a = [a_{1,1}, a_{2,1}, \ldots, a_{m,1}, a_{1,2}, a_{2,2}, \ldots, a_{m,2}, \ldots, a_{m,n}]^T, \tag{1.2.2}$$

while the inverse operator, mat, transforms a vector (1.2.2) to an associated matrix $A = [a_{i,j}] \in \mathbb{R}^{m\times n}$. Thus,

$$\mathsf{vec}(A) = a, \qquad \mathsf{mat}(a) = A.$$

Theorem 1.2.3. *The Kronecker product satisfies the following relations for matrices* A, B, C, D, X *of suitable sizes :*

$$\begin{aligned} (A \otimes B)\mathsf{vec}(X) &= \mathsf{vec}(BXA^T), \\ (A \otimes B)^T &= A^T \otimes B^T, \\ (AB) \otimes (CD) &= (A \otimes C)(B \otimes D). \end{aligned} \tag{1.2.3}$$

For matrices $A, B \in \mathbb{R}^{m \times n}$, we define the inner product

$$\langle A, B \rangle_F := \mathsf{tr}(A^T B), \tag{1.2.4}$$

where $\mathsf{tr}(\cdot)$ denotes the trace, and we note that

$$\langle A, B \rangle_F = (\mathsf{vec}(A))^T \mathsf{vec}(B).$$

The Frobenius norm is associated with this inner product,

$$\|A\|_F := \langle A, A \rangle_F^{1/2},$$

and satisfies

$$\|A\|_F = \|\mathsf{vec}(A)\|_2. \tag{1.2.5}$$

Two matrices $A, B \in \mathbb{R}^{m \times n}$ are said to be F-orthogonal if

$$\langle A, B \rangle_F = 0.$$

1.3 The $\diamond$ product [14]

Definition 1.3.1. Let $A = [A_1, A_2, ..., A_p]$ and $B = [B_1, B_2, ..., B_\ell]$ be matrices of dimension $n \times ps$ and $n \times \ell s$, respectively, where A_i and B_j $(i = 1, ..., p; j = 1, ..., \ell)$ are $n \times s$

matrices. Then the $p \times \ell$ matrix $A^T \diamond B$ is defined by

$$A^T \diamond B = \begin{bmatrix} \langle A_1, B_1 \rangle_F & \langle A_1, B_2 \rangle_F & \cdots & \langle A_1, B_\ell \rangle_F \\ \langle A_2, B_1 \rangle_F & \langle A_2, B_2 \rangle_F & \cdots & \langle A_1, B_\ell \rangle_F \\ \vdots & \vdots & & \vdots \\ \langle A_p, B_1 \rangle_F & \langle A_p, B_2 \rangle_F & \cdots & \langle A_p, B_\ell \rangle_F \end{bmatrix}. \tag{1.3.1}$$

Proposition 1.3.2. *Let A, B, $C \in \mathbb{R}^{n \times ps}$, and $D \in \mathbb{R}^{n \times n}$, $L \in \mathbb{R}^{p \times p}$. Then we have*

$$\begin{aligned} (A+B)^T \diamond C &= A^T \diamond C + B^T \diamond C, \\ A^T \diamond (B+C) &= A^T \diamond B + A^T \diamond C, \\ A^T \diamond (B\,(L \otimes I_s)) &= \left(A^T \diamond B\right) L. \end{aligned} \tag{1.3.2}$$

Proposition 1.3.3. *(The global QR factorization [14]). Let $Z = [Z_1, Z_2, ..., Z_k]$ be an $n \times ks$ matrix with $Z_i \in \mathbb{R}^{n \times s}$ for $i = 1, ..., k$. Then Z may be decomposed as*

$$Z = Q\,(R \otimes I_s)\,, \tag{1.3.3}$$

where $Q = [Q_1, ..., Q_k]$ is an $n \times ks$ F-orthonormal matrix satisfying $Q^T \diamond Q = I_k$ and R is an upper triangular $k \times k$ matrix.

Definition 1.3.4. (Hadamard product). The Hadamard product denoted by $\circ$ of two matrices, A, B, of the same dimension, is a matrix, with entries given by

$$(A \circ B)_{i,j} = (A)_{i,j}(B)_{i,j}. \tag{1.3.4}$$

Proposition 1.3.5. *The Hadamard product is commutative, associative and distributive over addition. That is,*

$$A \circ B = B \circ A, \tag{1.3.5}$$

$$A \circ (B \circ C) = (A \circ B) \circ C, \tag{1.3.6}$$

$$A \circ (B + C) = A \circ B + A \circ C \tag{1.3.7}$$

Chapitre 2

A global Lanczos method for Tikhonov regularization minimization problems with a solution norm constraint

2.1 Introduction

Image restoration is the process of removing blur and noise from an available degraded image to recover an approximation of the unavailable original noise- and blur-free image. Let the original image be represented by the matrix $\widehat{X} \in \mathbb{R}^{m\times n}$, whose entries represent pixels. The operator vec maps $\widehat{X}$ to the vector $\widehat{x} \in \mathbb{R}^{mn}$ defined by stacking the columns of $\widehat{X}$ from left to right. Blurring is a linear deterministic process which can be modeled by a matrix $H \in \mathbb{R}^{mn\times mn}$; see, e.g., [2, 6, 18, 62, 64] for discussions on image restoration. Thus, $\widehat{b} = H\widehat{x}$ represents a blurred image associated with $\widehat{x}$. The available image, represented by $b \in \mathbb{R}^{mn}$, is assumed to be contaminated by both blur and noise. Let the entries of the "noise vector" $e \in \mathbb{R}^{mn}$ be normally distributed with zero mean and variance δ^2. Then b

can be expressed as

$$b = H\widehat{x} + e. \tag{2.1.1}$$

The aim of image restoration is to determine an approximation of $\widehat{x}$ by computing an approximate solution of the linear system of equations

$$Hx = b. \tag{2.1.2}$$

If the system is inconsistent, then we consider it a least-squares problem. Blurring matrices H typically are very ill-conditioned and may be singular. Since b is contaminated by the error e, straightforward solution of (2.1.2) generally does not yield a useful approximation of $\widehat{x}$ due to severe propagation of the error e into the computed solution. Let $H^\dagger$ denote the Moore–Penrose pseudoinverse of H and let $\|\cdot\|_2$ be the Euclidean vector norm. Then the least-squares solution of minimal Euclidean norm of (2.1.2) is given by $H^\dagger b$ and, typically,

$$\|\widehat{x}\|_2 \ll \|H^\dagger b\|_2 \approx \|H^\dagger e\|_2.$$

This difficulty can be remedied by replacing (2.1.2) by a nearby problem, whose solution is less sensitive to the error in b. The replacement is referred to as regularization. One of the most popular regularization methods is due to Tikhonov [31, 49]. In its simplest form, Tikhonov regularization replaces the linear system (2.1.2) by the penalized least-squares problem

$$\min_{x\in\mathbb{R}^{mn}} \{\|Hx - b\|_2^2 + \mu\|x\|_2^2\}, \tag{2.1.3}$$

where $\mu > 0$ is a regularization parameter. The normal equations associated with (2.1.3) are given by

$$(H^T H + \mu I)x = H^T b, \tag{2.1.4}$$

where I is the identity matrix of suitable order and the superscript ^{T} denotes transposition. It follows that (2.1.3) has the unique solution

$$x_\mu := (H^T H + \mu I)^{-1} H^T b \tag{2.1.5}$$

for any $\mu > 0$. Our computed restoration of the vector b will be an approximation of a vector of the form (2.1.5).

Proposition 2.1.1. *Assume that $H^T b \neq 0$. Then $\|x_\mu\|_2$ is a monotonically decreasing function of μ and*

$$\lim_{\mu \searrow 0} \|x_\mu\|_2 = \|H^\dagger b\|_2, \qquad \lim_{\mu \to \infty} \|x_\mu\|_2 = 0. \tag{2.1.6}$$

Démonstration. The representation

$$\|x_\mu\|_2^2 = x_\mu^T x_\mu = b^T H (H^T H + \mu I)^{-2} H^T b \tag{2.1.7}$$

shows that $\|x_\mu\|_2$ is a decreasing function of $\mu > 0$. The right-hand side limit (2.1.6) is immediate ; the left-hand side limit follows by substituting the singular value decomposition of H into (2.1.7). □

The quality of the computed restoration depends on the choice of $\mu > 0$. Several approaches to choosing μ are described in the literature, including the discrepancy principle, generalized cross validation, and the L-curve ; see, e.g., [73, 90] for overviews and discussions. In this chapter, we will assume an estimate of $\|\widehat{x}\|_2$ to be available, and we will use this estimate to determine μ. Knowledge of the norm of the desired approximate solution of linear systems (2.1.2) is available in some applications, see, e.g., Ahmad et al. [1], and this approach to select μ is has received considerable attention in the literature ;

see, e.g., [24, 25, 91, 92]. It works well when the relative error $\|e\|_2/\|\widehat{b}\|_2$ is not too large; see [25] for illustrations.

The organization of this chapter is as follows. Section 2.2 describes how the Kronecker structure allows the application of the global Lanczos method for solving the normal equations (2.1.4). The section also shows how upper and lower bounds for $\|x_\mu\|_2^2$ can be determined by using the relation between the global Lanczos method and certain Gauss-type quadrature rules. This extends techniques developed by Golub and Meurant [46, 47] based on the relation between the standard Lanczos method and Gauss-type quadrature rules to the global Lanczos method. An algorithm for image restoration is presented in Section 2.3, and Section 2.4 describes a few computed examples. These examples illustrate the benefit of exploiting the Kronecker product structure of H.

2.2 The global Lanczos algorithm and Gauss quadrature

Let H_1 and H_2 be the Kronecker factors of the blurring matrix H and introduce the linear operator

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{m\times n} &\rightarrow \mathbb{R}^{m\times n} \\ \mathcal{A}(X) &= H_2 X H_1^T. \end{aligned}$$

Its transpose is given by $\mathcal{A}^T(X) = H_2^T X H_1$. Define the symmetric linear operator

$$\breve{\mathcal{A}}(X) = (\mathcal{A}^T \circ \mathcal{A})(X),$$

where $\circ$ denotes composition.

Proposition 2.2.1. *Let H have the Kronecker structure (1.2.1) and assume that $G := \mathcal{A}^T(B) \neq O$. Let $\mu > 0$. Then the equation*

$$(\breve{\mathcal{A}} + \mu I)(X) = G \tag{2.2.1}$$

has a unique solution $X_\mu \in \mathbb{R}^{m\times n}$. Let $b := \mathsf{vec}(B)$ and let x_μ be given by (2.1.5) with this vector b. Then

$$X_\mu = \mathsf{mat}(x_\mu). \tag{2.2.2}$$

Moreover, $\|X_\mu\|_F$ is a decreasing function of $\mu > 0$ with

$$\lim_{\mu\searrow 0} \|X_\mu\|_F = \|H^\dagger b\|_2, \qquad \lim_{\mu\to\infty} \|X_\mu\|_F = 0. \tag{2.2.3}$$

Démonstration. We have $\breve{\mathcal{A}}(X) = H_2^T H_2 X H_1^T H_1$. Therefore (2.2.1) can be written as

$$H_2^T H_2 X H_1^T H_1 + \mu X = H_2^T B H_1.$$

Using the properties (1.2.3), this equation can be expressed as

$$((H_1 \otimes H_2)^T (H_1 \otimes H_2) + \mu I)\mathsf{vec}(X) = (H_1 \otimes H_2)^T \mathsf{vec}(B),$$

which is the same as (2.1.4). This establishes (2.2.2). The properties of $\|X_\mu\|_F$ now follow from Proposition 2.1.1. □

The matrix $\widehat{X} = \mathsf{mat}(\widehat{x})$ represents the unavailable blur- and noise-free image that we would like to determine. We assume that an estimate of

$$\Delta := \|\widehat{X}\|_F \tag{2.2.4}$$

is known and will compute an approximation of $\widehat{X}$ by solving the constrained least-squares problem

$$\min_{\|X\|_F \leq \Delta} \|B - \mathcal{A}(X)\|_F. \tag{2.2.5}$$

The solution of this minimization problem without constraint is equivalent to solving (2.1.2) when H is nonsingular. The solution of the unconstrained problem typically is of very large norm due to contamination by propagated error ; see Proposition 2.2.2 below. If this solution has Frobenius norm larger than or equal to Δ, then the solution of (2.2.5) satisfies $\|X\|_F = \Delta$. We will assume this to be the case. This solution is described by the following proposition.

Proposition 2.2.2. *Assume that the solution of the minimization problem (2.2.5) without constraint is of norm larger than Δ. Then the solution of (2.2.5) is the solution of (2.2.1) for some $\mu > 0$.*

Démonstration. By (2.2.2) the solution X_μ of (2.2.1) is equivalent to the solution x_μ of (2.1.3) given by (2.1.5). A proof that x_μ is the unique solution of

$$\min_{\|x\|_2=\Delta} \|b - Hx\|_2 \tag{2.2.6}$$

for a suitable value of $\mu > 0$ can be established with the aid of Lagrange multipliers ; see Golub and von Matt [56]. It is easy to see that x_μ also is a solution when the constraint in (2.2.6) is replaced by $\|x\|_2 \leq \Delta$. The problem so obtained is equivalent to (2.2.5). □

We remark that if the solution of (2.2.5) without constraint has Frobenius norm smaller than or equal to Δ, then we may choose the regularization parameter $\mu = 0$. Let X_μ denote the solution of (2.2.1) for $\mu > 0$. Introduce the function

$$\phi(\mu) := \|X_\mu\|_F^2. \tag{2.2.7}$$

We will approximate ϕ to be able to determine an estimate of $\|X_\mu\|_F$ inexpensively. Our approximation is obtained by expressing (2.2.7) as a Stieltjes integral, and exploiting the

connection between the global Lanczos method and Gauss-type quadrature rules.

Proposition 2.2.3. *Introduce for $\mu > 0$ the function*

$$f_\mu(t) := (t + \mu)^{-2}. \tag{2.2.8}$$

Then (2.2.7) can be expressed as

$$\phi(\mu) = \int f_\mu(t) d\omega(t), \tag{2.2.9}$$

where $d\omega$ is a measure with support on the nonnegative real axis. Moreover, ϕ is decreasing and convex for $\mu > 0$ with

$$\lim_{\mu \searrow 0} \phi(\mu) = \|H^\dagger b\|_2^2, \qquad \lim_{\mu \to \infty} \phi(\mu) = 0. \tag{2.2.10}$$

Démonstration. It follows from (2.2.7), (2.2.2), (1.2.5), and (2.1.5) that

$$\phi(\mu) = \|x_\mu\|_2^2 = x_\mu^T x_\mu = bH^T(H^T H + \mu I)^{-2} H^T b. \tag{2.2.11}$$

Substituting the spectral factorization

$$H^T H = U \Lambda U^T,$$

where $\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_{mn}] \in \mathbb{R}^{mn \times mn}$ and $U \in \mathbb{R}^{mn \times mn}$ is orthogonal, into the right-hand side of (2.2.11), with $w = [w_1, w_2, \ldots, w_{mn}]^T := U^T H^T b$, gives

$$\phi(\mu) = \sum_{j=1}^{mn} f_\mu(\lambda_j) w_j^2.$$

The right-hand side is a Stieltjes integral, which can be expressed as (2.2.9). The distribution function ω associated with the measure $d\omega$ can be chosen as a nondecreasing piecewise constant function with nonnegative jumps w_j^2 at the eigenvalues λ_j. Since $H^T H$ is positive semidefinite, the support of the measure $d\omega$ lives on the nonnegative real axis.

It follows from (2.2.11) that the derivatives of ϕ satisfy $\phi'(\mu) < 0$ and $\phi''(\mu) > 0$, which shows that ϕ is decreasing and convex. The limits (2.2.10) are a consequence of (2.2.3). □

It is convenient to define the integral operator

$$\mathcal{I}f := \int f_\mu(t) d\omega(t) \tag{2.2.12}$$

and the associated inner product

$$[p, q] := \mathcal{I}(pq) \tag{2.2.13}$$

for polynomials p and q of low enough degree. We may define orthogonal polynomials with respect to this inner product and, therefore, also Gauss quadrature rules for the approximation of (2.2.12). The k-point Gauss quadrature rule $\mathcal{G}_k$ is characterized by the property that

$$\mathcal{G}_k p = \mathcal{I}p, \qquad \forall p \in \mathbb{P}_{2k-1},$$

where $\mathbb{P}_{2k-1}$ denotes the set of polynomials of degree at most $2k-1$. The remainder term for this quadrature rule can be expressed as

$$\mathcal{I}f_\mu - \mathcal{G}_k f_\mu = \frac{f_\mu^{(2k)}(\xi)}{(2k)!} \int \prod_{j=1}^{k} (t - t_j)^2 d\omega(t),$$

where $f_\mu^{(\ell)}$ denotes the derivative of order ℓ of the function (2.2.8), ξ is in the convex hull of the support of the measure $d\omega$, and $t_1, t_2, \ldots, t_k$ are the nodes of $\mathcal{G}_k$. Since $f_\mu^{(2k)}(\xi)$ is positive, it follows that $\mathcal{G}_k f < \mathcal{I}f$.

Let $\mathcal{R}_{k+1,0}$ denote the $(k+1)$-point Gauss–Radau rule for the measure $d\omega$ with a fixed node $t_0 = 0$. Then

$$\mathcal{R}_{k+1,0} p = \mathcal{I}p, \qquad \forall p \in \mathbb{P}_{2k}. \tag{2.2.14}$$

The quadrature error is

$$\mathcal{I}f_\mu - \mathcal{R}_{k+1,0}f_\mu = \frac{f_\mu^{(2k+1)}(\tilde{\xi})}{(2k+1)!}\int (t-t_0)\prod_{j=1}^{k}(t-\tilde{t}_j)^2 d\omega(t),$$

where $\tilde{\xi}$ is in the convex hull of the support of the measure $d\omega$ and the origin, and $\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_k$ are the "free" Gauss–Radau nodes; see, e.g., [46, 47]. In view of that $f_\mu^{(2k+1)}(\xi)$ is negative and the support of $d\omega$ lives on the nonnegative real axis, $\mathcal{R}_{k+1,0}f$ is an upper bound for $\mathcal{I}f$. Indeed, one can show that

$$\mathcal{G}_{k-1}f_\mu < \mathcal{G}_k f_\mu < \mathcal{I}f < \mathcal{R}_{k+1,0}f_\mu < \mathcal{R}_{k,0}f_\mu, \tag{2.2.15}$$

see, e.g., [79] for details.

We conclude that pairs of Gauss and Gauss–Radau quadrature rules $\mathcal{G}_k f_\mu$ and $\mathcal{R}_{k+1,0}f_\mu$ yield lower and upper bounds for $\phi(\mu)$. We will now describe how these quadrature rules can be evaluated by carrying out k steps of the global Lanczos method without explicit knowledge of the measure $d\omega$.

2.2.1 The global Lanczos method

The global Lanczos method can be applied to reduce a large symmetric matrix to a small tridiagonal one; see [67]. This method differs from the standard Lanczos method in that it uses the inner product (1.2.4) between matrices. This makes it possible to use matrix-matrix products during the execution of the global Lanczos method. These products can be evaluated efficiently on many computers. Algorithm 1 below executes k steps of the global Lanczos method applied to $\breve{\mathcal{A}}$ with initial matrix $V_1 = G/\|G\|_F$ of unit Frobenius norm with $G = \mathcal{A}^T(B)$.

Algorithm 1 The global Lanczos algorithm

Input : Let $\beta_1 = 0$, $V_0 = O$, and $V_1 = G/\|G\|_F$.

1. For $j = 1, 2, \ldots, k$

(a) $W = \breve{\mathcal{A}}(V_j) - \beta_j V_{j-1}$,

(b) $\alpha_j = \langle V_j, W \rangle_F$,

(c) $W = W - \alpha_j V_j$,

(d) $\beta_{j+1} = ||W||_F$,

(e) $V_{j+1} = W/\beta_{j+1}$,

3. EndFor

The zero matrix $V_0 \in \mathbb{R}^{m \times n}$ does not have to be stored. We assume that all coefficients β_{j+1} are positive. This is the generic situation. Otherwise, the algorithm breaks down, because the recursions cannot be continued. This is a very rare event. We therefore will not dwell on it further. Thus, under the assumption that all generated coefficients β_{j+1} are positive, the algorithm determines the matrices

$$\mathcal{V}_k = [V_1, V_2, \ldots, V_k] \in \mathbb{R}^{m \times kn}, \qquad \mathcal{V}_{k+1} = [V_1, V_2, \ldots, V_{k+1}] \in \mathbb{R}^{m \times (k+1)n},$$

whose "matrix columns" V_j are F-orthonormal, i.e.,

$$\langle V_i, V_j \rangle_F = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases} \tag{2.2.16}$$

It follows from the recursion formulas of Algorithm 1 and the definition of V_1 that

$$V_j = p_{j-1}(\breve{\mathcal{A}})(G), \qquad j = 1, 2, \ldots, k+1, \tag{2.2.17}$$

for some polynomials p_{j-1} of precisely degree $j-1$. We refer to these polynomials as global Lanczos polynomials. They satisfy the same recursion relations as the matrix columns

V_j ; see Section 2.2.2 below. Their explicit form is not required, only their existence. The property (2.2.17) together with the F-orthonormality (2.2.16) of the matrix columns shows that the set $\{V_j\}_{j=1}^{k+1}$ forms an F-orthonormal basis for the global Krylov subspace

$$\mathcal{K}_{k+1}(\breve{\mathcal{A}}, G) := \operatorname{span}\{G, \breve{\mathcal{A}}(G), \ldots, \breve{\mathcal{A}}^k(G)\},$$

where $\breve{\mathcal{A}}^j = \breve{\mathcal{A}}^{j-1} \circ \breve{\mathcal{A}}$ for $j = 2, 3, \ldots, k$.

The coefficients $\alpha_1, \alpha_2, \ldots, \alpha_k$ and $\beta_2, \beta_3, \ldots, \beta_k$ determined by Algorithm 1 define the symmetric tridiagonal matrix

$$T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \end{bmatrix} \in \mathbb{R}^{k\times k}, \tag{2.2.18}$$

and the recurrence formulas of Algorithm 1 can be expressed as

$$[\breve{\mathcal{A}}(V_1), \breve{\mathcal{A}}(V_2), \ldots, \breve{\mathcal{A}}(V_k)] = \mathcal{V}_k(I_n \otimes T_k) + \beta_{k+1}[O, \ldots, O, V_{k+1}], \tag{2.2.19}$$

where $O \in \mathbb{R}^{m\times n}$ denotes the zero matrix.

The scheme of this chapter is based on applying the global Lanczos method to $\breve{\mathcal{A}}$ (Algorithm 1). It is possible to develop an analogous scheme based on applying the global Lanczos bidiagonalization method to $\mathcal{A}$ and $\mathcal{A}^T$. A global Lanczos bidiagonalization method is described by Toutounian and Karimi [95]. The latter approach would give a method closely related to the scheme in [25], which is based on the standard Lanczos bidiagonalization method (also referred to as Golub–Kahan bidiagonalization). We apply the global Lanczos method because it requires less computer storage than the global

Lanczos bidiagonalization method for the same number of steps, since the latter generates two sets of F-orthonormal vectors while the former only determines one set. Since our aim is to develop a method suitable for large-scale problems, reducing the computer storage required is important.

2.2.2 Gauss-type rules associated with the global Lanczos method

We first show orthogonality of the polynomials p_{j-1} determined by (2.2.17). This provides the link between the global Lanczos method and Gauss quadrature.

Proposition 2.2.4. *The polynomials p_{j-1}, $j = 1, 2, \ldots$, defined by (2.2.17) are orthonormal with respect to the inner product (2.2.13).*

Démonstration. The definition of the measure $d\omega$ that determines the functional $\mathcal{I}$ in (2.2.13) yields

$$[p_{j-1}, p_{i-1}] = \int p_{j-1}(t)p_{i-1}(t)d\omega(t) = bH^T p_{i-1}(H^T H)p_{j-1}(H^T H)H^T b.$$

Using the linearity of $\breve{\mathcal{A}}$ and the properties (1.2.3) of the Kronecker product, we obtain that for any polynomial p,

$$\mathsf{vec}(p(\breve{\mathcal{A}})(G)) = p(H^T H)\mathsf{vec}(G).$$

Moreover, $\mathsf{vec}(G) = H^T b$. It follows that

$$\begin{aligned} [p_{j-1}, p_{i-1}] &= (\mathsf{vec}(p_{j-1}(\breve{\mathcal{A}})(G)))^T \mathsf{vec}(p_{i-1}(\breve{\mathcal{A}})(G)) \\ &= (\mathsf{vec}(V_j))^T \mathsf{vec}(V_i) = \langle V_j, V_i \rangle_F. \end{aligned}$$

The orthonormality of the polynomials p_{i-1} is a consequence of (2.2.16). □

It follows from Algorithm 1 or (2.2.19) that the V_j's satisfy the three-term recursion formula

$$\breve{\mathcal{A}}(V_j) = \alpha_j V_j + \beta_{j+1} V_{j+1} + \beta_j V_{j-1}, \qquad j = 1, 2, \ldots, k,$$

and by (2.2.17) the polynomials p_{j-1} satisfy the same recursions

$$t p_{j-1}(t) = \alpha_j p_{j-1}(t) + \beta_{j+1} p_j(t) + \beta_j p_{j-2}(t), \qquad j = 1, 2, \ldots, k.$$

These relations can be expressed as

$$\begin{aligned} [p_0(t), p_1(t), \ldots, p_{k-1}(t)] T_k &= t[p_0(t), p_1(t), \ldots, p_{k-1}(t)] \\ &- \beta_{k+1}[0, \ldots, 0, p_k(t)], \end{aligned}$$

which shows that the zeros of p_k are the eigenvalues of T_k. These zeros are the nodes of the k-point Gauss rule $\mathcal{G}_k$ associated with the measure $d\omega$. This Gauss rule applied to the function f_μ can be expressed as

$$\mathcal{G}_k f_\mu = ||G||_F^2 e_1^T f_\mu(T_k) e_1. \tag{2.2.20}$$

Throughout this chapter $e_j = [0, \ldots, 0, 1, 0, \ldots, 0]^T$ denotes the jth axis vector. The simple form of f_μ makes it possible to evaluate (2.2.20) without computing the spectral factorization of T_k. We will return to this below.

The relation between the standard Lanczos method applied to a symmetric matrix and Gauss quadrature has been exploited by Golub and Meurant [46, 47]; see also [21, 25]. The application of the relation between the global Lanczos method and Gauss quadrature to determine the regularization parameter in Tikhonov regularization generalizes the approach in [25] based on the standard Lanczos method.

The $(k+1)$-point Gauss–Radau quadrature rule $\mathcal{R}_{k+1,0}$ associated with the measure $d\omega$ and with a preassigned node $t_0 = 0$ can be represented analogously to (2.2.20). Define the matrix

$$T_{k+1,0} = \begin{bmatrix} T_k & \beta_{k+1}e_{k+1} \\ \beta_{k+1}e_k^T & \widetilde{t}_{k+1,k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times(k+1)}$$

by appending a row and a column to the matrix (2.2.18) to obtain a symmetric matrix with the last subdiagonal entry defined by (2.2.19) and the last diagonal entry to be determined. We would like this entry to be such that the matrix $T_{k+1,0}$ has the smallest eigenvalue zero. Note that since β_{k+1} is positive, the eigenvalues of T_k are in an open interval contained in the convex hull of the support of the measure $d\omega$. In particular, all eigenvalues of T_k are positive.

A simple way to determine $\widetilde{t}_{k+1,k+1}$ so that all eigenvalues of $T_{k+1,0}$ are nonnegative and one eigenvalue vanishes is to compute the bidiagonal Cholesky factor of $T_{k+1,0}$ (with $\widetilde{t}_{k+1,k+1}$ chosen large enough to make $T_{k+1,0}$ positive definite)

$$C_{k+1,0} = \begin{bmatrix} \rho_1 & & & & \\ \sigma_2 & \rho_2 & & & \\ & \ddots & \ddots & & \\ & & \sigma_k & \rho_k & \\ & & & \sigma_{k+1} & \rho_{k+1} \end{bmatrix}$$

and set $\rho_{k+1} = 0$. This determines the matrix $C_{k+1,0}$ and the desired matrix $T_{k+1,0}$ is given by

$$T_{k+1,0} = C_{k+1,0}C_{k+1,0}^T;$$

see, e.g., [21, Proposition 3.1] for a proof. Note that the entries $\rho_1, \rho_2, \ldots, \rho_k$ and $\sigma_2, \sigma_3, \ldots, \sigma_{k+1}$

of $C_{k+1,0}$ are independent of $\widetilde{t}_{k+1,k+1}$. The Gauss–Radau rule (2.2.14) can be evaluated according to

$$\mathcal{R}_{k+1,0} f_\mu = \|G\|_F^2 e_1^T f_\mu(T_{k+1,0}) e_1,$$

analogously to (2.2.20).

Introduce the functions

$$\phi_k^-(\mu) := \mathcal{G}_k f_\mu, \qquad \phi_{k+1}^+(\mu) := \mathcal{R}_{k+1,0} f_\mu. \tag{2.2.21}$$

Then, in view of (2.2.15),

$$\phi_{k-1}^-(\mu) < \phi_k^-(\mu) < \phi(\mu) < \phi_{k+1}^+(\mu) < \phi_k^+(\mu). \tag{2.2.22}$$

Instead of computing $\phi(\mu)$ for several values of μ, which is expensive for large-scale problems, we evaluate the upper and lower bounds (2.2.21) for desired values of μ. This is inexpensive when k is small. The computation of these bounds is discussed in the following subsection.

2.2.3 Evaluation of $\phi_k^-(\mu)$ and $\phi_{k+1}^+(\mu)$

We outline the evaluation of the functions $\phi_k^\mp(\mu)$ and the derivative of $\phi_{k+1}^+(\mu)$, which are needed when determining a suitable value of μ by our zero-finder. We have

$$\phi_k^-(\mu) = \|G\|_F^2 e_1^T (T_k + \mu I)^{-2} e_1.$$

For each value of $\mu > 0$, we solve the system of equations $(T_k + \mu I) z_\mu = e_1$ for $z_\mu \in \mathbb{R}^k$. Since the matrix $T_k + \mu I$ is symmetric and positive definite when $\mu > 0$, we solve this system with the aid of Cholesky factorization. Then

$$\phi_k^-(\mu) = \|G\|_F^2 z_\mu^T z_\mu.$$

Since $T_k + \mu I$ is tridiagonal, the evaluation of $\phi_k^-(\mu)$ requires only $\mathcal{O}(k)$ arithmetic floating point operations (flops) for each value of μ.

We turn to the computation of $\phi_{k+1}^+(\mu)$. The evaluation can be carried out similarly as the computation of $\phi_k^-(\mu)$ described above, with the matrix T_k replaced by $T_{k+1,0}$. However, since the Cholesky factor $C_{k+1,0}$ of $T_{k+1,0}$ is available, we outline an approach that uses this factor. We have

$$\phi_{k+1}^+(\mu) = \|G\|_F^2 e_1^T (C_{k+1,0} C_{k+1,0}^T + \mu I)^{-2} e_1. \tag{2.2.23}$$

Compute the solution of the least-squares problem

$$\min_{z \in \mathbb{R}^{k+1}} \left\| \begin{bmatrix} C_{k+1,0}^T \\ \mu^{1/2} I \end{bmatrix} z - \mu^{-1/2} e_{k+2} \right\|_2^2. \tag{2.2.24}$$

Denoting the solution by z_μ, we have

$$\phi_{k+1}^+(\mu) = \|G\|_F^2 z_\mu^T z_\mu.$$

The structure of the matrix in (2.2.24) makes it possible to compute z_μ by QR factorization of the matrix in only $\mathcal{O}(k)$ flops for each value of $\mu > 0$.

Changing the right-hand side in (2.2.24) gives the least-squares problem

$$\min_{s \in \mathbb{R}^{k+1}} \left\| \begin{bmatrix} C_{k+1,0}^T \\ \mu^{1/2} I \end{bmatrix} s - \mu^{-1/2} \begin{bmatrix} 0 \\ z_\mu \end{bmatrix} \right\|_2^2,$$

whose solution, s_μ, is used to evaluate the derivative

$$\frac{d}{d\mu} \phi_{k+1}^+(\mu) = -2 \|G\|_F^2 z_\mu^T s_\mu. \tag{2.2.25}$$

This also can be carried out in $\mathcal{O}(k)$ flops for each value of $\mu > 0$.

2.3 Computation of an approximate solution of specified norm

We describe a method for determining the regularization parameter μ for Tikhonov regularization (2.1.3). Specifically, we would like to determine $\mu > 0$ so that the computed solution X_μ satisfies

$$\min_{\|X\|_F=\Delta} \|B - \mathcal{A}(X)\|_F \tag{2.3.1}$$

with $\Delta > 0$ defined by (2.2.4). The constraint imposes that the restored image should have the same "energy" as the target image. Since this section is closely related to [25, Section 3], we only outline our approach and refer to [25] for further details.

Assume that $0 < \Delta < \|H^\dagger b\|_2$. Then, by Proposition 2.2.3, the equation

$$\phi(\mu) = \Delta^2 \tag{2.3.2}$$

has a unique solution $0 < \mu_\Delta < \infty$, which determines the desired solution X_{μ_Δ}. However, this equation is expensive to solve for large-scale problems, i.e., when mn is large. Similarly to the approach in [25], we therefore instead solve

$$\phi_{k+1}^+(\mu) = \Delta^2. \tag{2.3.3}$$

Proposition 2.3.1. *The function $\phi_{k+1}^+(\mu)$ defined by (2.2.21) is strictly decreasing and convex for $\mu > 0$. Equation (2.3.3) has a unique solution μ_{k+1} such that $0 < \mu_{k+1} < \infty$ for any positive finite Δ^2.*

Démonstration. The fact that ϕ_{k+1}^+ is strictly decreasing and convex follows from the representation (2.2.23), which also shows that $\lim_{\mu\to\infty} \phi_{k+1}^+(\mu) = 0$. Moreover, since all

subdiagonal entries of $T_{k+1,0}$ are nonvanishing, so do all subdiagonal entries of the Cholesky factor $C_{k+1,0}$. Therefore, by (2.2.23), $\lim_{\mu \searrow 0} \phi_{k+1}^+(\mu) = \infty$. □

Thus, whenever (2.3.2) has a unique bounded positive solution μ_Δ, then so does (2.3.3). We denote the solution of the latter by μ_{k+1}. Since both $\phi(\mu)$ and $\phi^+(\mu)$ are decreasing convex functions, it follows from (2.2.22) that

$$\mu_\Delta < \mu_{k+1} < \mu_k.$$

We apply the monotonically and quadratically convergent zero-finder described by Golub and von Matt [56, equations (75)–(78)] to compute approximations $\mu_{k+1}^{(j)}$, $j = 1, 2, \ldots$, of the zero μ_{k+1} of

$$h_{k+1}^+(\mu) := \phi_{k+1}^+(\mu) - \Delta^2. \tag{2.3.4}$$

This zero-finder requires repeated evaluation of the function (2.2.23) and its derivative (2.2.25).

We would like to compute a value of μ such that

$$\eta^2 \Delta^2 \leq \phi(\mu) \leq \Delta^2, \tag{2.3.5}$$

where the constant $0 \leq \eta \leq 1$ determines how close to Δ we require the norm of the computed approximate solution of (2.3.1) to be. We therefore calculate a value of μ such that

$$\eta^2 \Delta^2 \leq \phi_k^-(\mu), \qquad \phi_{k+1}^+(\mu) \leq \Delta^2. \tag{2.3.6}$$

This value satisfies (2.3.5). Using an initial approximation $\mu_{k+1}^{(0)}$ of the desired zero, μ_{k+1}, of (2.3.4), we first seek to satisfy the left-hand side inequality of (2.3.6). It may not

be possible to satisfy this inequality for small values of k. For instance, $\phi_k^-(\mu)$ may be negative for all $\mu > 0$ when k is small. We increase k by one when the left-hand side inequality of (2.3.6) cannot be satisfied, until k is large enough so that we can satisfy this inequality ; see [25] for details. Turning to the right-hand side inequality of (2.3.6), we use the zero-finder to determine $\mu_k^{(p)}$ such that

$$\frac{1}{10}(\eta^2 - 1)\Delta^2 + \Delta^2 \leq \phi_{k+1}^+(\mu_k^{(p)}) \leq \Delta^2.$$

In order for $\eta^2\Delta^2 \leq \phi_k^-(\mu_{k+1}^{(p)})$ to hold, we may have to increase k further ; see [25]. Having computed $\widehat{\mu} := \mu_k^{(p)}$, we determine the associated solution $\widehat{y}_k$ of $(T_k + \widehat{\mu}I)y = \|G\|_F e_1$ by solving the least-squares problem

$$\min_{y \in \mathbb{R}^{k+1}} \left\| \begin{bmatrix} C_k \\ \widehat{\mu}^{1/2} I \end{bmatrix} y - \widehat{\mu}^{-1/2} \|G\|_F e_{k+1} \right\|_2^2 .$$

Finally, our approximate solution of (2.3.1) is determined by

$$X_{\widehat{\mu},k} = \mathcal{V}_k(I \otimes \widehat{y}_k).$$

2.4 Numerical results

This section presents some representative numerical experiments. All computations were carried out using the MATLAB environment. The computations were carried out with approximately 15 decimal digits of accuracy. To secure that the global Lanczos block vectors V_j are numerically orthonormal, we reorthogonalize them with global QR factorization.

Algorithm 2 Outline of solution method for (2.3.1)

Input : Set $k = 2$ and let $V_1 := G/||G||_F$;

1. For $j = 1, 2, \ldots, k$

2. Determine the orthonormal basis $\{V_j\}_{j=1}^{k+1}$, the tridiagonal matrix T_k, and the last subdiagonal entry β_{k+1} of $T_{k+1,0}$ using Algorithm 1.

3. Determine an approximation of the zero $\widehat{\mu} = \mu_k$ of the function ϕ_{k+1}^{+} as described above. This may require k to be increased, in which case one returns to step 2.

4. Determine $\widehat{y}_k$ and $X_{\widehat{\mu},k}$ as described above.

Let $\widehat{x} = \mathsf{vec}(\widehat{X})$ denote the error-free exact solution of the linear system equation (2.1.2) and define

$$e = \mathsf{vec}(E), \qquad \widehat{b} = \mathsf{vec}(\widehat{B}), \qquad b = \mathsf{vec}(B),$$

where $\widehat{B} = H_2 \widehat{X} H_1^T$ and $B = \widehat{B} + E$. The error matrix E has normally distributed entries with zero mean and is normalized to correspond to a specific noise level

$$\nu = \frac{||E||_F}{||\widehat{B}||_F}.$$

To determine the effectiveness of our approach, we evaluate the relative error

$$\frac{||\widehat{X} - X_k||_F}{||\widehat{X}||_F}$$

of the computed approximate solution $X_k = X_{\widehat{\mu},k}$ of equation (2.2.1) obtained with Algorithm 2. The first two examples are concerned with the solution of Fredholm integral equations of the first kind. Discretization gives matrices that are severely ill-conditioned. The last two examples discuss image restoration problems.

Example 1

Let the nonsymmetric matrices H_1 and H_2, both of size 1500×1500, be determined by the MATLAB programs `baart` and `foxgood` in the Regularization Tools package by Hansen [61]. Specifically, we let $H_1 = \texttt{baart}(1500)$ and $H_2 = \texttt{foxgood}(1500)$. The computed condition numbers of these matrices are $\kappa(H_1) = 2 \times 10^{18}$ and $\kappa(H_2) = 3 \times 10^{13}$. Since $\kappa(H) = \kappa(H_1)\kappa(H_2)$, the matrix H is numerically singular. We determine the right-hand side matrix $\widehat{B}$ so that the exact solution is the matrix $\widehat{X}$ with all entries unity. Table 2.1 displays the computed regularization parameters and the relative error in approximate solutions determined by Algorithm 2 with $\eta = 0.99$ for different noise levels, as well as the number of iterations required to satisfy the stopping criterion.

TABLE 2.1 – Results for Example 1.

Noise level	Iterations	Regularization parameter	Relative error
0.001	8	2.0993×10^{-6}	8.54×10^{-2}
0.01	8	2.9595×10^{-6}	8.79×10^{-2}

Example 2

In this example, we consider the Fredholm integral equation

$$\int\int_{\Omega} K(x, y, s, t) f(s, t) ds dt = g(x, y), \qquad (x, y) \in \Omega', \tag{2.4.1}$$

where $\Omega = [0, \pi/2] \times [0, \pi/2]$ and $\Omega' = [0, \pi] \times [0, \pi]$. The kernel is given by

$$K(x, y, s, t) = k_1(x, s)k_1(y, t), \quad (x, y) \in \Omega', \quad (s, t) \in \Omega,$$

where

$$g(x,y) = g_1(x)g_1(y)$$

and

$$k_1(s,x) = \exp(\cos(x)), \quad g_1(s) = 2\sinh(s)/s.$$

We use the code `baart` from Regularization Tools [61] to discretize (2.4.1) by a Galerkin method with orthonormal box functions as test and trial functions to obtain H_1 of size 1500×1500, and let $H = H_1 \otimes H_1$. From the output of the code `baart` we determine a scaled approximation $\widehat{X} \in \mathbb{R}^{1500\times 1500}$ of the exact solution $f(t,s) = \sin(t)\sin(s)$. Table 2.2 shows the computed regularization parameters and the relative error in approximate solutions determined by Algorithm 2 with $\eta = 0.99$ and different noise levels, as well as the number of iterations required to satisfy the stopping criterion. Figure 2.1 displays the computed approximate solution X_8 obtained when the noise level ν of the available data (right-hand side) is 0.01.

TABLE 2.2 – Results for Example 2.

Noise level	Iterations	Regularization parameter	Relative error
0.001	8	9.4480×10^{-7}	4.57×10^{-2}
0.01	8	1.7235×10^{-6}	4.81×10^{-2}

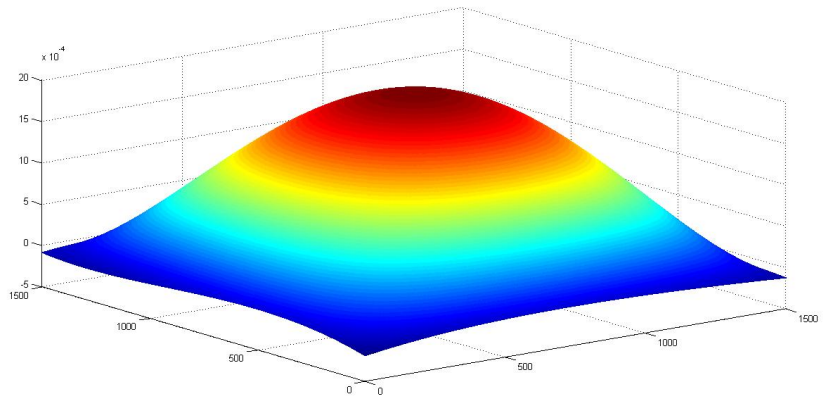


FIGURE 2.1 – Example 2 : Approximation X_8 of $\hat{X}$ determined by Algorithm 2 for noise level 0.01.

Example 3

We consider an image restoration problem. Then the vector $\widehat{b}$ in (0.0.2) represents the blurred image, H is a blurring matrix, $\widehat{x}$ represents the exact image and e is a noise vector. In general, the blurring matrix H is determined by the point spread function (PSF) [62], which defines how each pixel is blurred, and the boundary conditions, which specify our assumptions on the scene just outside our image. We assume that the horizontal and vertical components of the PSF can be separated. Then H can be expressed as a Kronecker product $H = H_1 \otimes H_2$; see [62]. The blurred image then is given by $H_2 \widehat{X} H_1^T$. Also when H cannot be written as a tensor product of two matrices, it may be possible to approximate H quite accurately by such a product. The factors can be determined by solving the minimization problem

$$\{\hat{H}_1, \hat{H}_2\} = \arg\min_{H_1, H_2} \| H - H_1 \otimes H_2 \|_F;$$

see [99].

The original image in this example is the `peppers` image of dimension 256×256 from MATLAB; it is shown in the left-hand side of Figure 2.2. The blurring matrix H is given by $H = H_1 \otimes H_2 \in \mathbb{R}^{256^2 \times 256^2}$, where $H_1 = H_2 = [h_{ij}]$ are Toeplitz matrices of dimension 256×256 with

$$h_{ij} = \begin{cases} \frac{1}{2r-1}, & |i-j| \leq r, \\ 0, & \text{otherwise.} \end{cases}$$

The blurring matrix H models uniform blur. In our example we set $r = 5$. The blurred and noisy image of Figure 2.2 is determined by adding noise of noise level $\nu = 0.001$ to the blurred image $H_2 \hat{X} H_1^T$. Table 2.3 summarizes the results obtained by Algorithm 2 with $\eta = 0.997$. The table also reports results obtained with the method proposed in [25]. This method utilizes the connection between (standard) Golub–Kahan bidiagonalization and Gauss quadrature rules for solving large ill-conditioned linear systems of equations (2.1.2). We refer to this method as GKB. It determines the regularization parameter analogously to Algorithm 2, and uses a similar stopping criterion, but does not exploit the structure of H. Table 2.3 shows Algorithm 2 to require less CPU-time (in seconds) than the GKB method in [25]. The norm of the error in the restored images determined by Algorithm 2 and the GKB method are about the same.

Example 4

The original image of this example is the `iograyBorder` image of dimension 256×256 from MATLAB; it is shown in the left-hand side of Figure 2.3. The blurring matrix is given by $H = H_1 \otimes H_2 \in \mathbb{R}^{256^2 \times 256^2}$, where H_1 and H_2 are the same matrices as in

TABLE 2.3 – Results for Example 3.

Method	Iterations	μ	Relative error	CPU time (seconds)
GKB	110	2.65×10^{-4}	6.79×10^{-2}	13.29
Algorithm 2	110	2.65×10^{-4}	6.79×10^{-2}	4.22

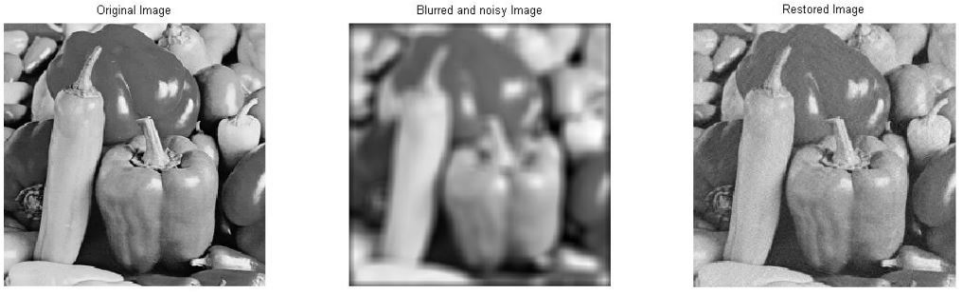


FIGURE 2.2 – Example 3 : Original image (left), degraded image (center), and restored image (right) for noise of level 0.001.

Example 3. In this example we set $r = 5$. The blurred and noisy image of Figure 2.3 is determined similarly as in Example 3 ; the noise level is 0.001. Table 2.4 summarizes the results obtained by Algorithm 2 with $\eta = 0.997$. The table also compares the number of iterations and the quality of the computed restored images obtained with Algorithm 2 and the GKB method. We can see that both methods yield restorations of the same quality and require about the same number of iterations.

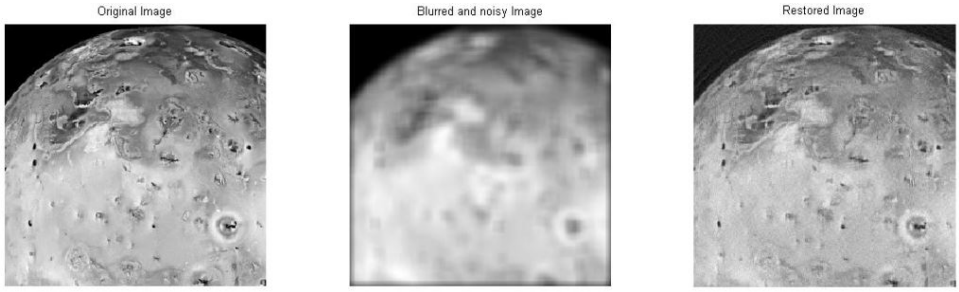


FIGURE 2.3 – Example 4 : Original image (left), degraded image (center), and restored image (right) for noise of level 0.001.

TABLE 2.4 – Results for Example 4.

Method	Iterations	μ	Relative error
GKB	76	2.89×10^{-4}	6.58×10^{-2}
Algorithm 2	75	2.90×10^{-4}	6.58×10^{-2}

Chapitre 3

Global Golub–Kahan bidiagonalization for Tikhonov regularization minimization problems with error norm constraint

3.1 Introduction

In this chapter we also consider the solution of large linear systems of equations (0.0.2) that arise from the discretization of ill-posed problems. The matrix has a Kronecker product structure and the right-hand side is contaminated by measurement error. Problems of this kind arise, for instance, from the discretization of Fredholm integral equations of the first kind in two space-dimensions with a separable kernel and in image restoration problems. Regularization methods, such as Tikhonov regularization, have to be employed to reduce the propagation of the error in the right-hand side into the computed solution. We investigate the use of the global Golub-Kahan bidiagonalization method to reduce the given large problem to a small one. The small problem is solved by employing Tikhonov regularization. A regularization parameter determines the amount of regularization.

The connection between global Golub-Kahan bidiagonalization and Gauss-type quadrature rules is exploited to inexpensively compute bounds that are useful for determining the regularization. We will assume the unavailable system of equations with error-free right-hand side,

$$Hx = \widehat{b}, \tag{3.1.1}$$

to be consistent and denote its solution of minimal Euclidean norm by $\widehat{x}$. It is our aim to determine an accurate approximation of $\widehat{x}$ by computing an approximate solution of the available linear system of equations (0.0.2). To reduce the effect of the noise in the reconstructed data, we will consider this time Tikhonov regularization in the following form

$$\min_{x\in\mathbb{R}^N}\{\|Hx - b\|_2^2 + \mu^{-1}\|x\|_2^2\}. \tag{3.1.2}$$

We will comment on the use of μ^{-1} instead of μ in (3.1.2) below. The minimization problem (3.1.2) has the unique solution

$$x_\mu := (H^T H + \mu^{-1} I)^{-1} H^T b. \tag{3.1.3}$$

The technique used for choosing a suitable value of μ in this chapter the *discrepancy principle*. For this method, when a bound ε of $\|e\|_2$ is available, the regularization parameter μ is commonly chosen so that

$$\|b - Hx_\mu\|_2 = \eta\varepsilon. \tag{3.1.4}$$

We would like to determine a value of μ such that

$$\varepsilon \leq \|b - Hx_\mu\|_2 \leq \eta\varepsilon, \tag{3.1.5}$$

where the constant $\eta > 1$ is independent of ε. This chapter continues our exploration of the application of global Krylov subspace methods to the solution of large-scale problems

(0.0.2) with a Kronecker structure that was begun in chapter 3. There a scheme for computing an approximation of $\widehat{x}$ of prescribed norm is described. It was convenient to base this scheme on the global Lanczos tridiagonalization method and use its connection to Gauss-type quadrature rules. This chapter focuses on the more common situation that a bound for the norm of the error e in b is available or can be estimated. Then the regularization parameter $\mu > 0$ can be determined by the discrepancy principle, i.e., so that the computed solution satisfies (3.1.5) ; see [31, 58]. The requirement (3.1.5) on the computed solution makes it natural to apply the GGKB method to develop an analogue of the approach in [26]. It is then the purpose of this chapter to develop an analogous method for the situation when H is the Kronecker product of two matrices H_1 and H_2. In applications of our solution method described in Section 3.5 both the matrices H_1 and H_2 are square. Then H is square. This simplifies the notation and, therefore, only this situation will be considered. However, only minor modifications of the method are necessary to handle the situation when one or both of the matrices H_1 and H_2 are rectangular.

This chapter is organized as follows. Section 3.2 discusses how the Kronecker product structure can be utilized when determining an approximate solution of (3.1.2) with the aid of the GGKB method. The connection between the GGKB method and Gauss-type quadrature rules is reviewed in Section 3.2, and the application of the GGKB method and Gauss-type quadrature to determine an approximate solution of (3.1.2) that satisfies (3.1.5) is described in Section 3.4. Numerical examples are presented in Section 3.5.

3.2 Gauss quadrature for Kronecker structure

Let H_1 and H_2 be Kronecker factors of the matrix H, and define the linear operator

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{m\times n} &\rightarrow \mathbb{R}^{m\times n} \\ \mathcal{A}(X) &:= H_2 X H_1^T. \end{aligned}$$

Its transpose is given by $\mathcal{A}^T(X) := H_2^T X H_1$. We will need the symmetric linear operator

$$\breve{\mathcal{A}}(X) := (\mathcal{A}^T \circ \mathcal{A})(X),$$

where $\circ$ denotes composition. It can be expressed as

$$\breve{\mathcal{A}}(X) = H_2^T H_2 X H_1^T H_1. \tag{3.2.1}$$

Let $B := \mathsf{mat}(b) \in \mathbb{R}^{m\times n}$ and assume that $G := \mathcal{A}^T(B) \neq O$, where $O \in \mathbb{R}^{m\times n}$ denotes the zero matrix. Assume that $0 < \mu < \infty$. Then the equation

$$(\breve{\mathcal{A}} + \mu^{-1} I)(X) = G \tag{3.2.2}$$

has a unique solution $X_\mu \in \mathbb{R}^{m\times n}$. Using (3.2.1) this equation can be written as

$$H_2^T H_2 X H_1^T H_1 + \mu^{-1} X = H_2^T B H_1.$$

With the aid of (1.2.3), we can express the above equation in the form

$$((H_1 \otimes H_2)^T (H_1 \otimes H_2) + \mu^{-1} I)\mathsf{vec}(X) = (H_1 \otimes H_2)^T \mathsf{vec}(B).$$

This equation is equivalent to the normal equations associated with (3.1.2). Therefore, $X_\mu = \mathsf{mat}(x_\mu)$.

Proposition 3.2.1. *Let $0 < \mu < \infty$ and let X_μ be the unique solution of* (3.2.2)*. Introduce the function*

$$\phi(\mu) := \|B - \mathcal{A}(X_\mu)\|_F^2. \tag{3.2.3}$$

Let $\eta > 1$ be the same as in (3.1.5)*. Then*

$$\varepsilon^2 \leq \phi(\mu) \leq \eta^2 \varepsilon^2 \tag{3.2.4}$$

is equivalent to that $x_\mu = \mathsf{vec}(X_\mu)$ satisfies (3.1.5) *with $b = \mathsf{vec}(B)$.*

Proof. Using (1.2.5) and (1.2.3), in order, yields

$$\phi(\mu) = \|B - H_2 X_\mu H_1^T\|_F^2 = \|\mathsf{vec}(B) - \mathsf{vec}(H_2 X_\mu H_1^T)\|_2^2 = \|b - (H_1 \otimes H_2) x_\mu\|_2^2,$$

and the proposition follows from $H = H_1 \otimes H_2$. □

The proposition implies that we can use X_μ instead of x_μ when determining a value of μ such that (3.1.5) holds. We will apply the following implementation of the GGKB method to determine an approximation of X_μ. The GGKB method generates two sequences of F-orthogonal matrices as well as a bidiagonal matrix.

The above algorithm determines the decompositions

$$\left.\begin{aligned} \mathcal{U}_{k+1}(\sigma_1 e_1 \otimes I_n) &= B, \\ \big[\mathcal{A}(V_1), \mathcal{A}(V_2), \ldots, \mathcal{A}(V_k)\big] &= \mathcal{U}_{k+1}(\bar{C}_k \otimes I_n), \\ \big[\mathcal{A}^T(U_1), \mathcal{A}^T(U_2), \ldots, \mathcal{A}^T(U_k)\big] &= \mathcal{V}_k(C_k^T \otimes I_n), \end{aligned}\right\} \tag{3.2.5}$$

where $e_j = [1, 0, \ldots, 0]^T$ denotes the first axis vector and $O \in \mathbb{R}^{m \times n}$ is a zero matrix. The matrices

$$\mathcal{V}_k = [V_1, V_2, \ldots, V_k] \in \mathbb{R}^{m \times kn}, \qquad \mathcal{U}_{k+1} = [U_1, U_2, \ldots, U_{k+1}] \in \mathbb{R}^{m \times (k+1)n},$$

Algorithm 3 The GGKB method

Input : Set $\sigma_1 = ||B||_F$, $U_1 = B/\sigma_1$, $V_1 = O$

1. For $j = 1, 2, \ldots, k$

(a) $\tilde{V}_j = \mathcal{A}^T(U_j) - \sigma_j V_j$

(b) $\rho_j = \|\tilde{V}_j\|_F$

(c) $V_{j+1} = \tilde{V}_j/\rho_j$

(d) $\tilde{U}_{j+1} = \mathcal{A}(V_j) - \rho_j U_j$

(e) $\sigma_{j+1} = ||\tilde{U}_{j+1}||_F$

(f) $U_{j+1} = \tilde{U}_{j+1}/\sigma_{j+1}$

3. EndFor

have F-orthonormal "matrix columns" $V_j \in \mathbb{R}^{m\times n}$ and $U_j \in \mathbb{R}^{m\times n}$, respectively, i.e.,

$$\langle V_i, V_j\rangle_F = \langle U_i, U_j\rangle_F = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$

Finally, the matrix $C_k \in \mathbb{R}^{k\times k}$ is bidiagonal,

$$C_k = \begin{bmatrix} \rho_1 & & & & \\ \sigma_2 & \rho_2 & & & \\ & \ddots & \ddots & & \\ & & \sigma_{k-1} & \rho_{k-1} & \\ & & & \sigma_k & \rho_k \end{bmatrix} \tag{3.2.6}$$

and so is

$$\bar{C}_k = \begin{bmatrix} C_k \\ \sigma_{k+1} e_k^T \end{bmatrix} \in \mathbb{R}^{(k+1)\times k}.$$

We remark that the matrix columns $U_1, U_2, U_3, \ldots$, are only required to advance the computations of Algorithm 3. Therefore, they all can be stored in the same memory location.

Also, the matrix column $\tilde{U}_{j+1}$ can use this location. The matrix columns $V_1, V_2, \ldots, V_k$ will be used to determine an approximate solution of (3.1.2) and, therefore, cannot be overwritten. However, the matrix columns $\tilde{V}_j$ and V_{j+1} may share the same storage location.

3.3 Gauss quadrature

This section discusses how the GGKB method is related to quadrature rules of Gauss-type. The connection follows via the relation between global Lanczos tridiagonalization and Gauss quadrature. Combining the equations (3.2.5) yields

$$[\breve{\mathcal{A}}(U_1), \breve{\mathcal{A}}(U_2), \ldots, \breve{\mathcal{A}}(U_k)] = \mathcal{U}_k(C_k C_k^T \otimes I_n) + \sigma_{k+1}[O, \ldots, O, U_{k+1}]. \tag{3.3.1}$$

This decomposition also can be determined by applying k steps of the global symmetric Lanczos (GSL) method to the matrix $\breve{\mathcal{A}}$ with initial block vector B. The GSL method is described in [67] and its relation to Gauss quadrature is discussed in [9]. We therefore only provide an outline. We remark that the relation between the standard symmetric Lanczos method and Gauss-type quadrature is well known; see, e.g., [26, 46, 47].

Proposition 3.3.1. *Assume that $H^T b \neq 0$. Then the function* (3.2.3) *can be expressed as*

$$\phi(\mu) = b^T(\mu H H^T + I)^{-2} b. \tag{3.3.2}$$

Proof. The proof of Proposition 3.2.1 shows that the function (3.2.3) can be written as

$$\phi(\mu) = \|b - H x_\mu\|_2^2.$$

Substituting (3.1.3) into the right-hand side and using the identity, for $\mu > 0$,

$$H(H^T H + \mu^{-1} I)^{-1} H^T = I - (\mu H H^T + I)^{-1} \tag{3.3.3}$$

gives (3.3.2). □

Substituting the spectral factorization $HH^T = W\Lambda W^T$, where $W \in \mathbb{R}^{mn \times mn}$ is orthogonal and $\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_{mn}] \in \mathbb{R}^{mn \times mn}$, into the right-hand side of (3.3.2) yields

$$\phi(\mu) = \sum_{j=1}^{mn} f_\mu(\lambda_j) w_j^2. \tag{3.3.4}$$

Here $[w_1, w_2, \ldots, w_{mn}]^T := W^T b$ and

$$f_\mu(t) := (\mu t + 1)^{-2}.$$

The expression (3.3.4) is a Stieltjes integral, which we write as

$$\phi(\mu) = \int f_\mu(t) d\omega(t). \tag{3.3.5}$$

The distribution function ω associated with the measure $d\omega$ can be chosen as a nondecreasing piecewise constant function with nonnegative jumps w_j^2 at the eigenvalues λ_j. Since HH^T is positive semidefinite, the support of the measure $d\omega$ lives on the nonnegative real axis.

Define the integral

$$\mathcal{I} f := \int f(t) d\omega(t)$$

for suitable functions f and let $\mathcal{G}_k$ denote the k-point Gauss quadrature rule associated with $d\omega$. It is characterized by the property that

$$\mathcal{G}_k p = \mathcal{I} p \qquad \forall p \in \mathbb{P}_{2k-1},$$

where $\mathbb{P}_{2k-1}$ denotes the set of polynomials of degree at most $2k-1$. The global Lanczos decomposition (3.3.1) provides a way to evaluate this Gauss rule without explicit knowledge of the measure $d\omega$. We have that for suitable functions f,

$$\mathcal{G}_k f = \|B\|_F^2 e_1^T f(T_k) e_1,$$

where the matrix $T_k := C_k C_k^T$ is symmetric and tridiagonal; see [9] for a proof. Analogous results that relate standard Golub–Kahan bidiagonalization to Gauss quadrature have been shown by Golub and Meurant [46, 47] and are applied in [26]. Gautschi [39] provides a nice fairly recent discussion on Gauss quadrature and orthogonal polynomials; see also [38].

We are particularly interested in the Gauss rule

$$\mathcal{G}_k f_\mu = \|B\|_F^2 e_1^T (\mu C_k C_k^T + I)^{-2} e_1. \tag{3.3.6}$$

Using the remainder formula for Gauss quadrature and the fact that all even-order derivatives of the integrand $t \to f_\mu(t)$ in (3.3.5) are positive, one can show that, generically,

$$\mathcal{G}_1 f_\mu < \cdots < \mathcal{G}_{k-1} f_\mu < \mathcal{G}_k f_\mu < \phi(\mu); \tag{3.3.7}$$

see, e.g., [9, 79] for details.

Let $\mathcal{R}_{k+1}$ denote the $(k+1)$-point Gauss–Radau rule for the measure $d\omega$ with a fixed node $t_0 = 0$. Then

$$\mathcal{R}_{k+1} p = \mathcal{I} p \qquad \forall p \in \mathbb{P}_{2k}.$$

This rule, when applied to the integration of f_μ, can be can expressed as

$$\mathcal{R}_{k+1} f_\mu = \|B\|_F^2 e_1^T (\mu \bar{C}_k \bar{C}_k^T + I)^{-2} e_1;$$

see [9]. The remainder formula for Gauss–Radau quadrature rules and the fact that all odd-order derivatives of $t \to f_\mu(t)$ are negative yield that, generically,

$$\phi(\mu) < \mathcal{R}_{k+1} f_\mu < \mathcal{R}_k f_\mu < \cdots < \mathcal{R}_1 f_\mu; \tag{3.3.8}$$

see, e.g., [9, 79].

We conclude that pairs of Gauss and Gauss–Radau quadrature rules $\mathcal{G}_k f_\mu$ and $\mathcal{R}_{k+1} f_\mu$ yield lower and upper bounds, respectively, for $\phi(\mu)$. We evaluate these rules by first executing Algorithm 3 and then solving the least-squares problem

$$\min_{z \in \mathbb{R}^k} \left\| \begin{bmatrix} \mu^{1/2} C_k^T \\ I_k \end{bmatrix} z - e_{k+1} \right\|_2^2. \tag{3.3.9}$$

The solution, denoted by z_k, satisfies

$$(\mu C_k C_k^T + I_k) z_k = e_1. \tag{3.3.10}$$

It follows from (3.3.6) that

$$\mathcal{G}_k f_\mu = \|B\|_F^2 z_k^T z_k.$$

The special structure of the least-squares problem (3.3.9) makes it possible to evaluate $\mathcal{G}_k f_\mu$ in only $\mathcal{O}(k)$ arithmetic floating-point operations for each value of μ. Typically $\mathcal{G}_k f_\mu$ has to be evaluated for several μ-values ; see Section 3.4. The evaluation of $\mathcal{R}_{k+1} f_\mu$ can be carried out analogously ; the matrix C_k^T in (3.3.9) has to be replaced by $\bar{C}_k^T$; see [9]. The reason for solving the least-squares problem instead of the associated normal equations (3.3.10) is that the solution of the former generally is less sensitive to errors in the data and to round-off errors introduced during the computations.

3.4 Parameter selection and computation of an approximate solution

This section describes how the bounds for $\phi(\mu)$ described in the previous section can be used to determine a suitable number of steps k of the GGKB method, a value μ_k of the regularization parameter, and an approximation $x_{\mu_k,k}$ of the vector x_{μ_k}, defined by (3.1.3) with $\mu = \mu_k$, that satisfies (3.1.5).

For a given value of $k \geq 2$, we solve the nonlinear equation

$$\mathcal{G}_k f_\mu = \varepsilon^2 \tag{3.4.1}$$

for μ. Because we use the parameter μ in (3.1.2), instead of $1/\mu$, the left-hand side is a decreasing convex function of μ. There is a unique solution, denoted by μ_ε, of

$$\phi(\mu) = \varepsilon^2 \tag{3.4.2}$$

for almost all values of $\varepsilon > 0$ of practical interest and therefore also of (3.4.1) for k sufficiently large; see [9, 26] for analyses. Various zero-finders can be applied, including Newton's method; see [78]. The evaluation of each iterate requires the solution of a least-squares problem (3.3.9). The following result shows that the regularization parameter determined by solving (3.4.1) provides more regularization than the parameter obtained by solving (3.4.2).

Proposition 3.4.1. *Let μ_k solve* (3.4.1) *and let μ_ε solve* (3.4.2)*. Then, generically, $\mu_k < \mu_\varepsilon$.*

Proof. It follows from (3.3.2) that $\phi(\mu)$ is a decreasing and convex function for $\mu \geq 0$

with $\phi(0) = \|b\|_2^2$. Similarly, by (3.3.6), $\mathcal{G}_k f_\mu$ is a decreasing and convex function for $\mu \geq 0$ with $\mathcal{G}_k f_0 = \|b\|_2^2$. Generically, $\phi(\mu) > \mathcal{G}_k f_\mu$ for $\mu > 0$; cf. (3.3.7). Therefore, typically, $\mu_k < \mu_\varepsilon$. We have equality in the rare event of breakdown of the recursion formulas for the GGKB method. Assume that the $\rho_k > 0$ in (3.2.6). Then the matrix (3.2.6) is nonsingular and the solution μ_k of (3.4.1) exists if $0 < \varepsilon < \|b\|_2$. Let $P_{\mathcal{N}(HH^T)}$ denote the orthogonal projector onto the null space of HH^T. Then the solution μ_ε of (3.4.2) exist if $\|P_{\mathcal{N}(HH^T)} b\|_2 < \varepsilon < \|b\|_2$. □

Having computed μ_k, we check whether

$$\mathcal{R}_{k+1} f_\mu \leq \eta^2 \varepsilon^2 \tag{3.4.3}$$

holds for $\mu = \mu_k$. If this is the case, then it follows from (3.3.7) and (3.3.8) that (3.2.4) is valid for $\mu = \mu_k$. If, on the other hand, the inequality (3.4.3) is violated for $\mu = \mu_k$, then we increase k by one, compute μ_{k+1}, and check whether (3.4.3) holds with $k+1$ replaced by $k+2$ for $\mu = \mu_{k+1}$. For most problems of interest, the Gauss and Gauss-Radau approximations (3.3.7) and (3.3.8) converge quite rapidly to $\phi(\mu_k)$ as k increases. Therefore, the bound (3.4.3) typically holds already for a fairly small value of k.

Assume that μ_k satisfies (3.4.1) and (3.4.3). Then we determine an approximate solution of (3.1.2) with the aid of the global Golub–Kahan decomposition (3.2.5) as follows. First we determine the solution y_{k,μ_k} of

$$(\bar{C}_k^T \bar{C}_k + \mu_k^{-1} I) y = \sigma_1 \bar{C}_k^T e_1, \qquad \sigma_1 = \|B\|_F. \tag{3.4.4}$$

It is computed by solving the least-squares problem

$$\min_{y \in \mathbb{R}^k} \left\| \begin{bmatrix} \mu_k^{1/2} \bar{C}_k \\ I \end{bmatrix}_k y - \sigma_1 \mu_k^{1/2} e_1 \right\|_2 .$$

Similarly as above, we solve this least-squares problem instead of the associated normal equations (3.4.4) because of the better numerical properties of the latter. Finally, our approximate solution of (3.2.2) is determined by

$$X_{k,\mu_k} = \mathcal{V}_k(y_{k,\mu_k} \otimes I). \tag{3.4.5}$$

Proposition 3.4.2. *The approximate solution* (3.4.5) *of* (3.2.2) *satisfies*

$$\varepsilon \leq \|B - \mathcal{A}(X_{k,\mu_k})\|_F \leq \eta\varepsilon. \tag{3.4.6}$$

Proof. Using the representation (3.4.5), and applying (3.2.5) as well as (1.2.3), shows that

$$\mathcal{A}(X_{k,\mu_k}) = \mathcal{U}_{k+1}(\bar{C}_k \otimes I_n)(y_{k,\mu_k} \otimes I_n) = \mathcal{U}_{k+1}(\bar{C}_k y_{k,\mu_k} \otimes I_n).$$

Substituting the above expression into (3.4.6) and again using (3.2.5) yields

$$\begin{aligned} & \|\mathcal{U}_{k+1}(\sigma_1 e_1 \otimes I_n) - \mathcal{U}_{k+1}(\bar{C}_k y_{k,\mu_k} \otimes I_n)\|_F^2 \\ = \;& \|(\sigma_1 e_1 \otimes I_n) - (\bar{C}_k y_{k,\mu_k} \otimes I_n)\|_F^2 \\ = \;& \|\sigma_1 e_1 - \bar{C}_k y_{k,\mu_k}\|_2^2, \end{aligned}$$

where we recall that $\sigma_1 = \|B\|_F$. We now express y_{k,μ_k} with the aid of (3.4.4), and apply the identity (3.3.3) with H replaced by $\bar{C}_k$, to obtain

$$\begin{aligned} & \|B - \mathcal{A}(X_{k,\mu_k})\|_F^2 \\ = \;& \sigma_1^2 \|e_1 - \bar{C}_k(\bar{C}_k^T \bar{C}_k + \mu_k^{-1} I_k)^{-1} \bar{C}_k^T e_1\|_2^2 \\ = \;& \sigma_1^2 e_1^T (\mu_k \bar{C}_k \bar{C}_k^T + I_{k+1})^{-2} e_1 \\ = \;& \mathcal{R}_{k+1} f_{\mu_k}. \end{aligned}$$

The proposition now follows from (3.4.3) and the fact that $\varepsilon^2 = \mathcal{G}_k f_{\mu_k} \leq \mathcal{R}_{k+1} f_{\mu_k}$. □

Algorithm 4 The GGKB-Tikhonov method

Input : H_1, H_2, B, ε, $\eta \geq 1$

1. Set $k = 2$ (k is the number of global Lanczos steps). Let $U_1 := B/||B||_F$;

2. Determine the orthonormal bases $\{U_j\}_{j=1}^{k}$ and $\{V_j\}_{j=1}^{k}$, and the bidiagonal matrices C_k and $\bar{C}_k$ with Algorithm 2.

3. Determine μ_k that satisfies (3.4.1) and (3.4.3) as described above. This may require k to be increased, in which case one returns to step 2.

4. Determine y_{k,μ_k} and X_{k,μ_k} as described above.

The following algorithm summarizes the computation of (3.4.5).

We now comment on the complexity of Algorithm 4. We first mention that the overall computational cost for Algorithm 4 is dominated by the work required to determine U_j and V_j at step 2. The computational effort required to determine U_j and V_j is dominated by four matrix-matrix products, approximately $4(m+n)N$ arithmetic floating point operations (flops). These matrix-matrix operations need less computational cost in comparison with the cost required for evaluating matrix-vector evaluation with the large matrix H and its transpose, that is to say approximately $4N^2$ flop counts.

3.5 Numerical examples

This section presents a few representative numerical experiments. Let $\widehat{x} := \mathsf{vec}(\widehat{X})$ denote the error-free exact solution of the linear system of equations (0.0.2), let $\widehat{B} := H_2\widehat{X}H_1^T$ and $B := \widehat{B} + E$, and define

$$\widehat{b} := \mathsf{vec}(\widehat{B}), \qquad b := \mathsf{vec}(B), \qquad e := \mathsf{vec}(E),$$

where the error matrix E has normally distributed entries with zero mean and is normalized to correspond to a specific noise level

$$\nu := \frac{||E||_F}{||\widehat{B}||_F}.$$

The sizes of the matrices is specified in the examples below.

To determine the effectiveness of our solution method, we evaluate the relative error

$$\frac{||\widehat{X} - X_k||_F}{||\widehat{X}||_F}$$

of the computed approximate solution $X_k = X_{k,\mu_k}$ determined by Algorithm 4. The first three examples are concerned with the solution of Fredholm integral equations of the first kind in two space-dimensions with a separable kernel. Discretization gives matrices that are severely ill-conditioned. The last two examples discuss image restoration problems. Overviews of image restoration problems can be found in [18, 62].

Example 1.

Let the nonsymmetric matrix H_1 and the symmetric matrix H_2, both of size 1500×1500, be determined by the MATLAB programs `baart` and `foxgood`, respectively, in the Regularization Tools package by Hansen [61]. These programs provide discretizations of Fredholm integral equations of the first kind described in [3, 33]. The computed condition numbers of these matrices are $\kappa(H_1) = 3.72 \times 10^{19}$ and $\kappa(H_2) = 1.03 \times 10^{21}$. Since $\kappa(H) = \kappa(H_1)\kappa(H_2)$, the matrix H is numerically singular. The right-hand side matrix $\widehat{B}$ and the exact solution $\widehat{X}$ also are generated with the aid of the codes `baart` and `foxgood`. Table 3.1 displays the computed regularization parameters and the relative error in computed approximate solutions determined by Algorithm 4 with $\eta = 1.1$ for

different noise levels, as well as the number of iterations required to satisfy the stopping criterion.

TABLE 3.1 – Results for Example 1.

Noise level	Iterations	μ	Relative error
0.01	4	5.77×10^2	2.08×10^{-1}
0.001	7	2.64×10^4	1.22×10^{-1}

Example 2.

Instead of determining an approximate solution of (0.0.2) with Algorithm 4, one could compute the singular value decompositions (SVDs) of the matrices H_1 and H_2. The SVD of the matrix H then can be determined from the SVDs of H_1 and H_2. We can apply the SVD of H to determine a suitable regularization parameter $\mu > 0$. Specifically, we solve the equation $\|b - Hx_\mu\|_2 = \eta\varepsilon$ for μ by Newton's method without explicitly forming the matrices in the SVD of H. Knowing μ allows us to compute the Tikhonov solution (3.1.3) without explicitly forming the matrices in the SVD of H. This approach of determining the regularization parameter and computing the corresponding regularized solution (3.1.3) is attractive due to its simplicity when the matrices H_1 and H_2 are small. However, the approach is slow for large matrices H_1 and H_2. To illustrate this, we solve the same problem as in Example 1 with a finer discretization. Thus, let $H_1, H_2 \in \mathbb{R}^{2000\times 2000}$ be determined by the MATLAB codes `baart` and `foxgood`, respectively, from [58]. The matrix $\widehat{B}$ and $\widehat{X}$ are generated in the same manner as in Example 1 with the matrix B such

that the noise level is $\nu = 0.01$. We let the safety factor for the discrepancy principle be $\eta = 1.1$. Then Algorithm 4 terminates after 4 iterations with the regularization parameter $\mu_4 = 5.77 \times 10^2$ and an approximate solution, defined by (3.4.5) with $k = 4$, with relative error 2.09×10^{-1}. The computing time for this experiment is 37.96 seconds.

When we instead use the SVD of H, we obtain the regularization parameter $\mu = 22.8$ and an approximate solution with relative error 2.14×10^{-1}. The computing time for this method is 47.95 seconds. The difference in the values of the regularization parameter for these approaches depends on that in Algorithm 4 the regularization parameter is determined for the solution in a subspace of low dimension, while the solution determined by the SVD approach lives in the whole space.

The computations of this example illustrate that the computations with the global Golub–Kahan method are faster than using the SVD of H, even when the structure of the latter is exploited. The difference in computing time is even more pronounced for larger problems. We remark that also for problems of the size considered in Example 1, the global Golub–Kahan method is faster than using the SVD of H.

Example 3.

We consider the Fredholm integral equation

$$\int \int_{\Omega} K(x, y, s, t) f(s, t) ds dt = g(x, y), \qquad (x, y) \in \Omega, \tag{3.5.1}$$

where $\Omega = [-\pi/2, \pi/2] \times [-\pi/2, \pi/2]$. The kernel is given by

$$K(x, y, s, t) = k(x, s)k(y, t), \quad (x, y), (s, t) \in \Omega,$$

where

$$k(s,x) = (\cos(s) + \cos(x))^2(\sin(\xi)/\xi)^2, \quad \xi = \pi(\sin(s) + \sin(x)).$$

The solution is the sum of two Gaussians in each space dimension.

We discretize (3.5.1) by a Nyström method based on the composite trapezoidal rule with equidistant nodes in each space-dimension. Code is available at [84]. Specifically, we determine the nonsymmetric matrix $H_1 \in \mathbb{R}^{1500\times 1500}$, from which we obtain the discretization $H = H_1 \otimes H_1$ of the integral operator (3.5.1), as well as the exact solution $\widehat{X} \in \mathbb{R}^{1500\times 1500}$ of the discretized problem and the associated exact right-hand side $\widehat{B}$. Table 3.2 shows the computed regularization parameters and the relative error in approximate solutions determined by Algorithm 4 with $\eta = 1.01$ and different noise levels, as well as the number of iterations required to satisfy the stopping criterion. Figure 3.1 displays the computed approximate solution X_{13} obtained when the noise level of the available data (right-hand side) is 0.01.

TABLE 3.2 – Results for Example 3.

Noise level	Iterations	μ	Relative error
0.01	13	2.46×10^2	1.59×10^{-1}
0.001	32	2.67×10^4	6.97×10^{-2}

Example 4.

This and the following examples are concerned with the restoration of images that have been contaminated by blur and noise. Let the entries of the vector $\widehat{x}$ be pixel values for a

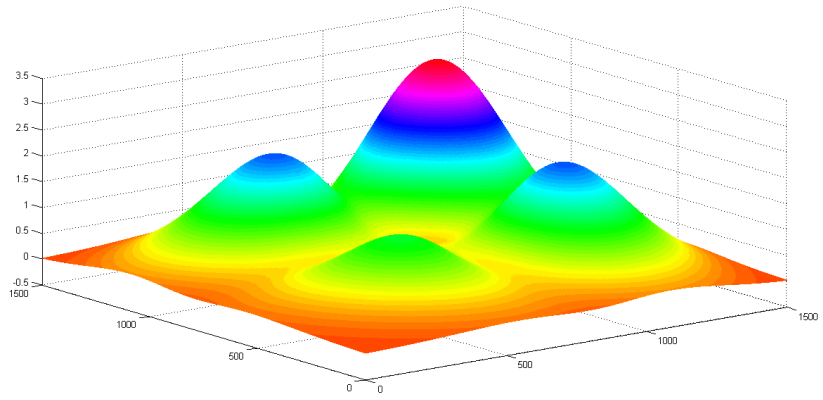


FIGURE 3.1 – Example 3 : Approximation X_{13} of $\hat{X}$ determined by Algorithm 4 for noise level 0.01.

desired, but unknown, image. The matrix H is a discretization of a blurring operator and equation (3.1.1) shows that $\widehat{b}$ represents a blurred, but noise-free, image. The vector b in (0.0.2) represents the available blur and noise contaminated image associated with $\widehat{x}$. The blurring matrix H is determined by the point-spread function (PSF), which determines how each pixel is smeared out (blurred), and by the boundary conditions, which specify our assumptions on the scene just outside the available image ; see [18, 62] for details.

In some cases the horizontal and vertical components of the PSF can be written as a product of two functions, one depending on the horizontal coordinate and the other one on the vertical coordinate, only. In this situation, the blurring matrix H can be expressed as a Kronecker product $H = H_1 \otimes H_2$. Let the matrix $\widehat{X} = \mathsf{mat}(\widehat{x})$ be of suitable size. Then the blurred image can be represented as $H_2 \widehat{X} H_1^T$; cf. (1.2.3). The blur and noise contaminated image is represented by the matrix $H_2 \widehat{X} H_1^T + E$, where E is the noise matrix. Also when H cannot be written as a Kronecker product of two matrices, it may

be possible to approximate H well by such a Kronecker product ; see [71, 72, 99].

In this example, we seek to restore an image that has been contaminated by blur that is defined by a Gaussian PSF,

$$h_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\big(-\frac{x^2+y^2}{2\sigma^2}\big),$$

and by noise. The Dirichlet zero boundary condition is imposed. The blurring matrix then is the Kronecker product of a symmetric Toeplitz matrices with itself $H = H_1 \otimes H_1$, where $H_1 = [h_{ij}]$ with

$$h_{ij} = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(i-j)^2}{2\sigma^2}\right), & |i-j| \le r, \\ 0 & \text{else} \end{cases}$$

The matrix H models atmospheric turbulence blur. We let $\sigma = 2.5$ and $r = 6$.

The original image $\widehat{X} \in \mathbb{R}^{256\times 256}$ is the `Enamel` image from MATLAB. It is shown in the left-hand side of Figure 3.2. The associated blurred and noisy image $B := H_2\widehat{X}H_1^T + E$ is shown in the middle of Figure 3.2 ; the noise level is 0.001. The restoration determined by Algorithm 4 with $\eta = 1.1$ is displayed in the right-hand side of the figure.

For comparison, we determine a regularization parameter and an approximate solution using the numerical method described in [26]. This method uses (standard) Golub–Kahan bidiagonalization instead of global Golub–Kahan bidiagonalization, and explores the connection between (standard) Golub–Kahan bidiagonalization and Gauss quadrature rules for solving large ill-conditioned linear systems of equations (0.0.2) without exploiting the structure of the matrix H. We refer to this method as GKB in Table 3.3. The table compares results obtained by Algorithm 4 and GKB, including the relative errors of the restorations, the number of iterations, and the CPU times required for two noise levels.

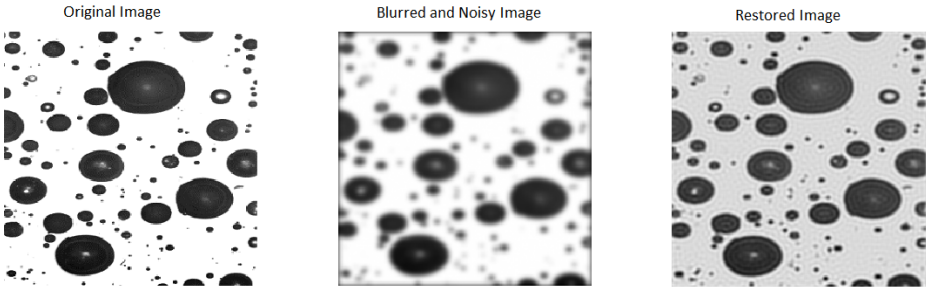


FIGURE 3.2 – Example 4 : Original image (left), degraded image (center), and restored image (right) for noise of level 0.001.

Algorithm 4 is seen to require less CPU time than GKB and give about the same quality of the computed restoration as GKB.

TABLE 3.3 – Results for Example 4.

Noise level	Iterations	Method	μ	Relative error	CPU-time (sec)
0.01	14	Algorithm 4	4.66×10^3	1.02×10^{-1}	0.50
	14	GKB	4.66×10^3	1.02×10^{-1}	1.35
0.001	62	Algorithm 4	1.71×10^4	8.00×10^{-2}	2.23
	62	GKB	1.71×10^4	8.02×10^{-2}	11.44

Example 5.

The original image is the `iograyBorder` image of dimension 256×256 from MATLAB. It is shown on the left-hand side of Figure 3.3. The blurring matrix $H = H_1 \otimes H_2$ is the same as in Example 3. The blurred and noisy image shown in the middle of Figure 3.3

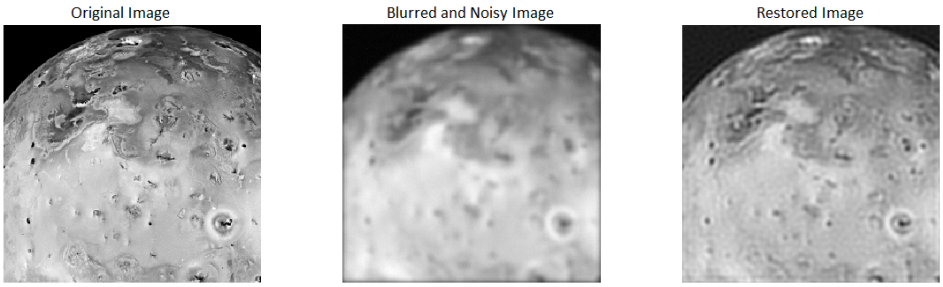


FIGURE 3.3 – Example 5 : Original image (left), degraded image (center), and restored image (right) for noise of level 0.01.

has the noise level 0.01. The restored image determined with Algorithm 4 with $\eta = 1$ is shown in the right-hand side of Figure 3.3. The number of iterations, relative errors of the restored images, and computed regularization parameters are shown in Table 3.4 for two noise levels.

TABLE 3.4 – Results for Example 5.

Noise level	Iterations	μ	Relative error	CPU time (sec)
0.01	19	4.80×10^4	8.85×10^{-2}	1.09
0.001	75	2.60×10^4	7.58×10^{-2}	2.13

Chapitre 4

The symmetric block Lanczos algorithm for linear ill-posed problems

4.1 Introduction

Many applications require the solution of several ill-conditioning systems of equations with a right hand side contaminated by an additive error,

$$Hx^{(i)} = \widehat{b}^{(i)} + e^{(i)}, \quad i = 1, 2, ..., k. \tag{4.1.1}$$

In recent decades, many iterative methods have been proposed for the solution of linear ill-posed problems. These iterative methods are based on the connection between the classical Lanczos algorithm, and Gauss quadrature formulas, for the solution of Tikhonov regularization of large-scale problems $Hx = b$, $H \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$. With a matrix H of ill-determined rank. The right hand side b is contaminated by an error $e \in \mathbb{R}^n$; see, e.g., [21, 22, 46, 47, 25, 26]. These techniques have been developed by Golub and Meurant in [46, 47] for approximating elements of a matrix function by Gaussian quadrature in the spectral domain. When solving (6.3.9), instead of applying these techniques to each ill-

posed linear system, it is more efficient to use a method for all the systems simultaneously. Consider the block linear system of equations,

$$HX = B, \quad \widehat{B} + E, \tag{4.1.2}$$

where H is an $n \times n$ real non singular matrix. X, $\widehat{B}$ and E are $n \times k$ rectangular matrices whose columns are $x^{(1)}, x^{(2)}, ..., x^{(k)}, \widehat{b}^{(1)}, \widehat{b}^{(2)}, ..., \widehat{b}^{(k)}$ and $e^{(1)}, e^{(2)}, ..., e^{(k)}$, respectively. Generalization of the standard methods utilizing the connection between classical Lanczos algorithm, orthogonal polynomials, and Gauss-type quadrature rules have been developed. The first class of these methods is based on the block Lanczos methods coinciding with Gauss-type quadrature rules and anti-Gauss quadrature rules for the symmetric and nonsymmetric block Lanczos algorithms; see, for example, [34, 46, 47] and references therein. The second class is based on the global Lanczos algorithms, with connection to Gauss-type quadrature rules [9]. In [51, 52, 67, 95] and references therein, a class of methods, based on block Krylov subspace methods, was introduced for the purpose of solving the following unavailable error-free linear system of equations with multiple right hand sides,

$$HX = \widehat{B}. \tag{4.1.3}$$

In the present chapter, we propose a new numerical method, based on the connection between the block Lanczos algorithm, block Gauss quadrature and anti-Gauss quadrature, for Tikhonov regularization of large linear system of equations with multiple right hand sides, where the right hand side matrix is contaminated by an error. Following [9, 25], we will show how an estimate of the norm of the exact solution can lead to estimate the value of the regularization parameter and the corresponding regularized solution. The error E is

absorbed in the right hand side B of (4.1.2), which makes the linear system of equations of multiple right hand sides (4.1.3) unavailable. We would like to determine the least-squares solution $\widehat{X}$ of minimal Frobenius norm of the unavailable error-free linear system of equations with multiple right hand sides, (4.1.3), which assumed to be consistent, by computing an approximate solution of the available linear system of equations (4.1.2). Since the coefficient matrix H is ill-conditioned, the problem will be extremely sensitive to perturbations in the right hand side matrix. In order to diminish the effect of the noise in the data, we replace the original coefficient matrix by a better conditioned one. One of the most popular regularization methods is due to Tikhonov [31, 49]. The method replaces the problem by the new one

$$X_\mu = \underset{X}{\operatorname{argmin}} \left(\|HX - B\|_F^2 + \mu \|LX\|_F^2 \right), \tag{4.1.4}$$

where $\mu \geq 0$ is the regularization parameter and L is a regularization matrix chosen to obtain a solution with desirable properties. The matrix L could be the identity matrix or a discrete form of first or second derivative. In the first case, the parameter μ acts on the size of the solution, while in the second case μ acts on the smoothness of the solution. We will only consider the particular case where the matrix L reduces to the identity I. Therefore, Tikhonov regularization problem in this case is of the following form

$$X_\mu = \underset{X}{\operatorname{argmin}} \left(\|HX - B\|_F^2 + \mu \|X\|_F^2 \right). \tag{4.1.5}$$

Tikhonov regularization problems of the form (4.1.4) can be brought into the form (4.1.5) ; see [11, 32]. In Tikhonov regularization, the solution of the problem (4.1.5) is computed

as the unique solution of the following linear least squares problem

$$X_\mu = \underset{X}{\operatorname{argmin}} \left\| \begin{bmatrix} H \\ \mu^{1/2} I \end{bmatrix} X - \begin{bmatrix} B \\ 0 \end{bmatrix} \right\|_F^2, \tag{4.1.6}$$

in certain situations the best way to solve (4.1.6) numerically is to treat the normal equations

$$\left(H^T H + \mu I\right) X = H^T B, \tag{4.1.7}$$

for a suitable positive value of the regularization parameter μ. The choice of μ affects how sensitive X_μ is to the error E in the contaminated right-hand side, and how accurately X_μ approximates $\widehat{X}$. Many techniques for choosing a suitable value of μ have been analyzed and illustrated in the literature; see, e.g., [17, 31, 58, 73, 90] and references therein. We remark that for any $\mu \geq 0$, the problem (4.1.7) has the unique solution

$$X_\mu = \left(H^T H + \mu I\right)^{-1} H^T B. \tag{4.1.8}$$

The condition number of the matrix $H^T H + \mu I$ is a decreasing function of μ and therefore the solution X_μ is generally more sensitive to the error E, the smaller $\mu \geq 0$ is. We will show how this problem can be remedied by implicitly solving the normal equation without working on $H^T H$.

In this chapter, we assume that the quantity

$$\Delta := \|\hat{X}\|_F,$$

is available and therefore we can use an equivalent formulation of problem (4.1.5), as a quadratically constrained least squares problem

$$\min_{\|X\|_F \leq \Delta} \| B - HX \|_F; \tag{4.1.9}$$

see, e.g., [9, Proposition 2.2] for further details. In the optimization literature, minimization problems of the form (4.1.9) arise in the context of trust-region methods.

The organization of this chapter is as follows. Section 4.2 reviews some known facts about block Gauss quadrature rules, block anti-gauss quadrature and their connection with the symmetric block Lanczos process. Section 4.3 describes how these connection can be employed to inexpensively compute an approximate regularizing parameter of μ by solving a nonlinear equation and then computing the corresponding approximate solution of X. In section 4.4 we discuss some numerical examples that demonstrate the validity of our approach and compare our proposed method to some related works.

4.2 Block Lanczos algorithm, block Gauss quadrature, block anti-Gauss quadrature and matrix functionals

Introduce,

$$\phi(\mu) := \|X_\mu\|_F^2. \tag{4.2.1}$$

In this chapter we review some results on the connection between the symmetric block Lanczos tridiagonalization, block Gauss quadrature rules, and block ani-Gauss quadrature rules, then we show how we can exploit this connection to inexpensively determine an approximation of ϕ. By substituting (4.1.8) into (4.2.1), the function (4.2.1) can be expressed as

$$\phi(\mu) = \operatorname{tr}\left(B^T H(H^T H + \mu I)^{-2} H^T B\right).$$

Introduce the function

$$f_\mu(t) := (t+\mu)^{-2}, \tag{4.2.2}$$

the expression (4.2.1) can be now expressed as $\mathrm{tr}(S)$, where

$$S := B^T H f_\mu \left(H^T H\right) H^T B. \tag{4.2.3}$$

We provide now an overview of the techniques that will be used to approximate (4.2.3). We first proceed to write the quantitie S as Stieltjes integral. Introduce the spectral factorization $H^T H = W \Lambda W^T$, where $W \in \mathbb{R}^{n\times n}$ satisfies $W^T W = I_n$ and $\Lambda = \mathrm{diag}[\lambda_1, ..., \lambda_n]$, with $\lambda_1 \leq ... \leq \lambda_n$. Defining $\Upsilon = [\Upsilon_1, ..., \Upsilon_n] = B^T H Q \in \mathbb{R}^{k\times n}$, where $\Upsilon_i \in \mathbb{R}^k$, it follows that

$$S = \Upsilon f_\mu(\Lambda)\Upsilon^T = \sum_{i=1}^{n} f_\mu(\lambda_i)\Upsilon_i\Upsilon_i^T = \int f_\mu(\lambda) d\Upsilon(\lambda) := \mathcal{I} f_\mu, \tag{4.2.4}$$

where $\Upsilon : \mathbb{R} \to \mathbb{R}^{k\times k}$ is discrete matrix distribution with a jump of size $\Upsilon_i\Upsilon_i^T$ at each eigenvalue λ_i of $H^T H$.

The quantity S has to be evaluated for several μ-values. This can be expensive, since in real applications the matrix H is large. However, the computation of low-dimensional approximation of S for several values of μ can be carried out efficiently by using block Gauss quadrature rules and block anti-Gauss quadrature. We briefly review some known facts about block Gauss quadrature rules and block anti-Gauss quadrature and their connection with block Lanczos process; see [34] for discussion on these methods. We define for a real λ, the following $k \times k$ orthogonal polynomial,

$$p_i(\lambda) = \sum_{j=0}^{i} \lambda^j C_j^{(i)},$$

where the coefficients $C_j^{(i)}$ are given $k \times k$ real matrices. We also define the following inner product induced by the measure $\Upsilon(\lambda)$, which is supposed to be symmetric and positive

semi-definite,

$$< p, q >= \int p(\lambda) d\Upsilon(\lambda) q(\lambda)^T,$$

and let $p_i, i = 1, 2, ...$ be a sequence of matrix polynomials orthonormal with respect to this inner product, i.e.,

$$\langle p_i, p_j \rangle = \delta_{ij} I_k,$$

where δ_{ij} is the Kronecker symbol and I_k is the identity matrix of order k.

The sequence of matrix orthonormal polynomials p_i satisfy a block three-term recurrence of the form

$$\lambda p_{j-1}(\lambda) = p_j(\lambda)\Gamma_j + p_{j-1}(\lambda)\Omega_j + p_{j-2}(\lambda)\Gamma_{j-1}^T, \quad p_0(\lambda) := I_k, \quad p_{-1}(\lambda) := 0.$$

For each j, the recursion coefficients Γ_j and Ω_j are $k \times k$ matrices with real entries. Morever, Ω_j is symmetric, and Γ_j can be chosen to be upper triangular. Defining

$$P_l(\lambda) := [p_0(\lambda), ..., p_\ell(\lambda)] \in \mathbb{R}^{k \times k\ell},$$

it follows that

$$\lambda P_\ell(\lambda) = P_\ell(\lambda) J_{kl} + P_\ell(\lambda)\Gamma_\ell E_\ell^T, \tag{4.2.5}$$

where

$$J_{k\ell} := \begin{bmatrix} \Omega_1 & \Gamma_1^T & & & \\ \Gamma_1 & \Omega_2 & \Gamma_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{\ell-2} & \Omega_{\ell-1} & \Gamma_{\ell-1}^T \\ & & & \Gamma_{\ell-1} & \Omega_\ell \end{bmatrix} \in \mathbb{R}^{k\ell \times k\ell}.$$

Throughout this chapter, $E_i := [e_{(i-1)k+1}, ..., e_{ik}]$ denotes the block vector of appropriate dimensions with blocks of size $k \times k$, with the ith block equal to I_k and all other blocks

equal to 0. Here, $J_{k\ell}$ is both a symmetric block-tridiagonal matrix and a banded matrix of bandwidth $2k+1$. The matrix $J_{k\ell}$ is computed via a partial block Lanczos tridiagonalization of the matrix H^TH without explicit knowledge of the measure measure $d\Upsilon$. In the following subsection, we will use the block Lanczos algorithm, which is first introduced for symmetric matrices, to reduce the symmetric matrix H^TH to a small symmetric block-tridiagonal matrix $J_{k\ell}$ and we will show how this algorithm is applied implicitly to H^TH, by applying block Lanczos bidiagonalization to H, which also known as the Golub-Kahan bidiagonalization.

4.2.1 Partial block Lanczos of the matrix H^TH

Algorithm 5 Partial block Lanczos process of the matrix H^TH

1. Let $P_1 \in \mathbb{R}^{n\times k}$ be an initial matrix satisfying $P_1^TP_1 = I_k$
2. Set $P_0 := 0 \in \mathbb{R}^{n\times k}$, $\Gamma_0 = 0 \in \mathbb{R}^{k\times k}$
3. For $j = 1, 2, \ldots, \ell$
   (a) $\Omega_j = P_j^TH^THP_j$
   (b) $R_j = H^THP_j - P_j\Omega_j - P_{j-1}\Gamma_{j-1}^T$
   (c) $P_{j+1}\Gamma_j = R_j$ (QR factorization), P_{j+1} is orthogonal and Γ_j is upper triangular
4. EndFor

Let $H^TB = QR$ be the QR factorization of H^TB, in the context of estimating (4.2.3), we choose $P_1 = Q$. The block-size k must be optimized considering the matrix size and the hardware capabilities, while the number of the partial block Lanczos steps ℓ depends upon the convergence speed of the algorithm on that particular problem. We will show

in our numerical results how this algorithm can be executed so that $\ell k \ll n$. The partial block Lanczos algorithm will break down at step j if R_j is rank deficient. In this case, the algorithm can be continued by replacing linearly dependent columns of P_{j+1} that have been orthogonalized with respect to all previous Lanczos vector ; see, e.g., [4] for further details. Breakdowns are very rare events, we therefore will not dwell on it further and assume that the blocks produced by the block Lanczos process are of full rank ; see, e.g., [55, Section 9.2.6] for a discussion on block Lanczos tridiagonalization. If the algorithm does not break down before step ℓ, then it is easy to verify the following relation

$$H^T H P_\ell^{(k)} = P_\ell^{(k)} J_{k\ell} + P_{\ell+1} \Gamma_\ell E_\ell^T, \tag{4.2.6}$$

where $P_\ell^{(k)} = [P_1, ..., P_\ell]$, and $J_{k\ell}$ is the matrix from (4.2.5). Moreover, the vector -columns $P_\ell^{(k)}$ form an orthonormal basis of the block Krylov subspace

$$\mathbb{K}_\ell(H^T H, P_1) = \text{Range}[P_1, H^T H P_1, (H^T H)^2 P_1, ..., (H^T H)^{\ell-1} P_1].$$

We now describe how the symmetric block-tridiagonal matrix $J_{k\ell}$ can be computed via the partial block Lanczos bidiagonalization to H, which also known as the block Golub-Kahan bidiagonalization. The latter method replaces matrix-block $H^T H X$ evaluations in the partial block Lanczos algorithm by $H^T X$ and HX operations. Application of ℓ steps of the block Lanczos bidiagonalization algorithm, given by Golub, Luk, and Overton in [45] to the matrix H with the initial block P_1 with orthonormal columns yields the decompositions

$$H P_\ell^{(k)} = Q_\ell^{(k)} R_{k\ell} \tag{4.2.7}$$

$$H^T Q_\ell^{(k)} = P_\ell^{(k)} R_{k\ell}^T + F_k E_k^T, \tag{4.2.8}$$

where $P_\ell^{(k)} \in \mathbb{R}^{n\times \ell k}$, $Q_\ell^{(k)} = [Q_1, ..., Q_\ell] \in \mathbb{R}^{n\times \ell k}$, ${P_\ell^{(k)}}^T P_\ell^{(k)} = {Q_l^{(k)}}^T Q_l^{(k)} = I_{\ell k}$. We refer to $F_k \in \mathbb{R}^{n\times k}$ as the residual matrix. It satisfies

$$ {P_\ell^{(k)}}^T F_k = 0. $$

The matrix

$$ R_{k\ell} := \begin{bmatrix} S_1 & L_2 & & & \\ & S_2 & L_2 & & \\ & & S_3 & \ddots & \\ & & & \ddots & L_\ell \\ & & & & S_\ell \end{bmatrix} \in \mathbb{R}^{k\ell\times k\ell}, \tag{4.2.9} $$

is upper block-bidiagonal with upper triangular diagonal blocks $S_j \in \mathbb{R}^{k\times k}$ and lower triangular superdiagonal blocks $L_j \in \mathbb{R}^{k\times k}$. Thus, $R_{k\ell}$ is upper triangular. We refer to the decompositions (4.2.7)-(4.2.8) as a partial block Lanczos bidiagonalization of H. The number of block-bidiagonalization steps, ℓ, is assumed to be small enough so that the decompositions (4.2.7)-(4.2.8) with the stated properties exist. The matrix (4.2.9) is assumed to be sufficiently small to allow easy computation of the regularization parameter μ. We will use the connection between partial block Lanczos bidiagonalization (4.2.7)-(4.2.8) of H and partial block Lanczos tridiagonalization of the matrix $H^T H$. Multiplying equation (4.2.7) by H^T from the left-hand side yields

$$ H^T H P_\ell^{(k)} = P_\ell^{(k)} R_{k\ell}^T R_{k\ell} + F_k E_k^T R_{k\ell}. \tag{4.2.10} $$

The symmetric and block-tridiagonal matrix $T_{k\ell} = R_{k\ell}^T R_{k\ell} \in \mathbb{R}^{k\ell\times k\ell}$ coincides with the matrix $J_{k\ell}$ in (4.2.6), and the expression (5.2.7) is then a partial block Lanczos tridiagonalization of $H^T H$ with initial block-vector P_1.

4.2.2 Block Gauss quadrature

The integral $\int f_\mu(\lambda)d\Upsilon(\lambda)$ is $k \times k$ symmetric matrix. We introduce the following ℓ-block Gauss quadrature rules associated with the matrix distribution Υ. The most general quadrature formula is of the form

$$\int f_\mu(\lambda)d\Upsilon(\lambda) = \sum_{i=1}^{\ell} W_i f_\mu(T_i) W_i^T, \tag{4.2.11}$$

where W_i and T_i are symmetric $k \times k$ matrices. By diagonalizing each T_i , we can obtain simpler formula. Let $T_i = Q_i \Lambda_i Q_i^T$ where Q_i is the orthonormal matrix of the eigenvectors, and Λ_i the diagonal matrix of the eigenvalues of T_i. This gives

$$\int f_\mu(\lambda)d\Upsilon(\lambda) = \sum_{i=1}^{\ell} W_i Q_i f_\mu(\Lambda_i) Q_i^T W_i^T$$

(4.2.11) can be reduced to

$$\sum_{i=1}^{k} f_\mu(t_i) u_i u_i^T,$$

where t_i is scalar and u_i is a vector with k components. Since our discussion is based on the connexion between Gauss quadratures and block Lanczos algorithm, it is then convenient to look at these quadratures in terms of the matrix $J_{k\ell}$. We therefore introduce the spectral factorization of $J_{k\ell}$,

$$J_{k\ell} = Y_{k\ell} \Theta_{k\ell} Y_{k\ell}^T,$$

where

$$Y_{k\ell} = [y_1^{(\ell)}, ..., y_{k\ell}^{(\ell)}], \qquad \Theta_{k\ell} = diag[\theta_1^{(\ell)}, ..., \theta_{k\ell}^{(\ell)}],$$

Y_ℓ satisfies $Y_\ell^T Y_\ell = I_\ell$, and $\theta_1^{(\ell)} \leq ... \leq \theta_{k\ell}^{(\ell)}$. From (4.2.5), the nodes t_i are the zeros of the determinant of the matrix orthogonal polynomials that is the eigenvalues $\theta_i^{(\ell)}$ of J_ℓ and

u_i is the vector consisting of the k first components of the corresponding eigenvector $y_i^{(\ell)}$. We then consider the following quadrature formula,

$$\mathcal{G}_\ell f_\mu := \sum_{i=1}^{k\ell} f(\theta_i^{(\ell)}) u_i^{(\ell)} \left(u_i^{(\ell)}\right)^T.$$

For the integral (4.2.4), we have the following property,

$$\mathcal{G}_\ell p = \mathcal{I} p,$$

for any polynomials of degree not exceeding $2\ell - 1$; see [34, Theorem 5] for a proof. The remainder term for this quadrature rule, which is a $k \times k$ matrix, can be expressed as

$$\mathcal{I} f_\mu - \mathcal{G}_\ell f_\mu = \frac{f_\mu^{(2\ell)}(\eta)}{(2\ell)!} \int s(\lambda) d\Upsilon(\lambda), \tag{4.2.12}$$

where $\eta \in [\theta_1, \theta_\ell]$ and $s(x) = (x - \theta_1)...(x - \theta_{2l\ell})$; see e.g., [47]. We call $\mathcal{G}_\ell f_\mu$ the ℓ-block Gauss quadrature rule associated with the matrix distribution Υ. Unfortunately, the factors of s do not have a constant sign over the interval $[\lambda_1, \lambda_n]$ and therefore the remainder formula is of unknown sign over this interval. We will see how this problem can be solved by the use of anti-Gauss quadrature in order to provide a quadrature formula whose error is of opposite sign with respect to the error of a block Gauss rule. It is useful to note that the ℓ-block Gauss rule affords the elegant and concise matrix

$$\begin{aligned}
\mathcal{G}_\ell f_\mu &= \sum_{i=1}^{k\ell} f_\mu(\theta_i^{(\ell)}) u_i^{(\ell)} \left(u_i^{(\ell)}\right)^T, \\
&= \left[u_1^{(\ell)}, ..., u_{k\ell}^{(\ell)}\right] f_\mu(\Theta_{k\ell}) \left[u_1^{(\ell)}, ..., u_{k\ell}^{(\ell)}\right]^T, \\
&= E_1^T Y_\ell f_\mu(\Theta_{k\ell}) Y_\ell^T E_1, \\
&= E_1^T f_\mu(J_{k\ell}) E_1.
\end{aligned} \tag{4.2.13}$$

Therefore, we can evaluate the Gauss quadrature rule $\mathcal{G}_\ell f_\mu$ without forming the spectral factorization of $J_{k\ell}$

4.2.3 Block anti-Gauss quadrature

In [34], block anti-Gauss quadrature rules computed via the symmetric block Lanczos algorithm were used to compute an approximate bound for $\mathcal{I}f_\mu$ with $k > 1$ extending the results introduced by Laurie in [74]. Let $\mathcal{H}_{\ell+1}f_\mu$ denote the $(\ell+1)$-block anti Gauss quadrature rule for the measure Υ. The block anti-Gauss quadrature is a tool to estimate the error in Gauss quadrature when no useful information with regard to the sign of the quadrature error can be obtained from the remainder formula. The idea is to construct a quadrature rule whose error is equal but of opposite sign to the error of block Gauss rule (4.2.12). The $(\ell+1)$-block anti-Gauss quadrature rule $\mathcal{H}_{\ell+1}f_\mu$ associated with the Gauss rule $\mathcal{G}_\ell f_\mu$ is characterized by

$$(\mathcal{I} - \mathcal{H}_{\ell+1})f = -(\mathcal{I} - \mathcal{G}_\ell)f, \quad f \in \mathbb{P}^{2\ell+1}. \tag{4.2.14}$$

where $\mathbb{P}^{2\ell+1}$ denotes the set of polynomials of degree at most $2\ell+1$. Note that (4.2.14) is exactly the $(\ell+1)$-block Gauss quadrature rule with respect to the matrix-valued function $2\mathcal{I} - \mathcal{G}_\ell$, since (4.2.14) implies that

$$\mathcal{H}_{\ell+1}f = (2\mathcal{I} - \mathcal{G}_\ell)f, \quad f \in \mathbb{P}^{2\ell+1}.$$

With respect to $2\mathcal{I} - \mathcal{G}_\ell$, we can show that there exists a sequence of polynomials $\tilde{p}_j$ with $k \times k$ matrix coefficients, orthonormal with respect to the inner product (4.2), satisfying the following three-term recurrence relation of the form

$$\lambda\tilde{p}_{j-1}(\lambda) = \tilde{p}_j(\lambda)\tilde{\Gamma}_j + \tilde{p}_{j-1}(\lambda)\tilde{\Omega}_j + \tilde{p}_{j-2}(\lambda)\tilde{\Gamma}_{j-1}^T, \quad \tilde{p}_0(\lambda) := I_k, \quad \tilde{p}_{-1}(\lambda) := 0.$$

As above, the $(\ell+1)$-block anti-Gauss quadrature rule affords the matrix representation

$$\mathcal{H}_{\ell+1}f_\mu = E_1^T f_\mu(\tilde{J}_{k(\ell+1)})E_1 \tag{4.2.15}$$

where,

$$\tilde{J}_{k(\ell+1)} := \begin{bmatrix} \tilde{\Omega}_1 & \tilde{\Gamma}_1^T & & & \\ \tilde{\Gamma}_1 & \tilde{\Omega}_2 & \tilde{\Gamma}_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde{\Gamma}_{\ell-1} & \tilde{\Omega}_\ell & \tilde{\Gamma}_\ell^T \\ & & & \tilde{\Gamma}_\ell & \tilde{\Omega}_{\ell+1} \end{bmatrix} \in \mathbb{R}^{k(\ell+1)\times k(\ell+1)},$$

with almost no work, from the matrix $J_{k(\ell+1)}$ associated with the $(\ell+1)$-block Gauss quadrature rule $\mathcal{G}_{\ell+1} f_\mu$, we obtain the matrix $\tilde{J}_{k(\ell+1)}$ associated with the $(\ell+1)$-block anti-Gauss quadrature rule $\mathcal{H}_{\ell+1} f_\mu$,

$$\begin{aligned} \tilde{\Omega}_i &= \Omega_i, \quad 1 \le i \le \ell \\ \tilde{\Gamma}_i &= \Gamma_i, \quad 1 \le i \le \ell - 1 \\ \tilde{\Gamma}_\ell &= \sqrt{2}\Gamma_\ell, \\ \tilde{\Omega}_{\ell+1} &= \Omega_{\ell+1}; \end{aligned}$$

see [34] for further details.

4.3 Computation of an approximate solution of specified norm

The quadratically constrained least squares formulation (4.1.9) has the advantage that, in some applications, the physical properties of the problem determine an optimal value for the norm constraint Δ, where Δ represents the energy of the exact solution. Knowledge of the norm of the desired approximate solution of ill-posed linear system of type (4.1.2) is available in some applications, see e.g, Ahmad et al. [1]. This approach

works well to determine an optimal value of the regularization parameter when the physical properties given by the norm constraint Δ is available. This is the case, for example in image restoration when the norm constraint represents the energy of the target image. This idea to select the regularization parameter is considered attractive to many authors and has received considerable attention in the literature; see, e.g, [9, 24, 25, 91, 92]. Determining the regularization parameter μ_Δ can be then achieved by solving the following nonlinear equation,

$$\phi(\mu) = \Delta^2, \tag{4.3.1}$$

which also determines the corresponding regularized solution X_{μ_Δ}. The computation of a μ-value such that (4.3.1) holds generally requires the use of a zero-finding method and typically $\phi(\mu)$ has to be evaluated for several μ-values. This can be expensive when the matrix H is large. In general the available value of Δ is an estimate of the norm of the exact solution. Therefore it is not meaningful to solve equation (4.3.1) to very high accuracy. Instead we will seek to determine a value of μ_Δ, such that

$$\eta^2\Delta^2 \leq \phi(\mu) \leq \Delta^2$$

where the constant $0 \leq \eta \leq 1$ determines how close to Δ we require the norm of the computed approximate solution of (4.1.9) to be. The more accurate the estimate is, the closer we choose η be to one. We exploit the connection between block-Gauss, block anti-Gauss quadrature rules, and block Lanczos algorithms, to obtain an average integration rule for $\mathcal{I}f$. For each ℓ, approximations of $[\mathcal{I}f]_{ij}$ are given by $[\mathcal{G}_\ell]_{ij}$ and $[\mathcal{H}_{\ell+1}]_{ij}$, $1 \leq i, j \leq k$. The purpose of our proposed numerical method is to keep the number of block

Lanczos algorithm steps ℓ small. Formula (4.2.14) suggests that the quadrature rule,

$$\mathcal{L}_{2l+1} f_\mu := \frac{1}{2}(\mathcal{G}_\ell f_\mu + \mathcal{H}_{\ell+1} f_\mu), \tag{4.3.2}$$

yields an approximation of $\mathcal{I} f_\mu$ at every step ℓ. Recall that the quadrature rule (4.3.2) is exact for polynomials of degree $2\ell + 1$. In order for $\mathcal{L}_{2\ell+1} f_\mu$ to be a good approximation of $\mathcal{I} f_\mu$, the following stopping criteria is used,

$$E_{\ell,\mu} := \frac{1}{2}\|G_\ell f_\mu - H_{\ell+1} f_\mu\|_{\max} < \tau \tag{4.3.3}$$

where τ is an absolute tolerance and $\|B\|_{\max} = \max_{1\le i,j\le k}|B_{ij}|$. We therefore start with $\ell = 1$ and an initial guess $\mu_\ell^{(0)}$ to determine approximations $\mu_\ell^{(j)}$, $j = 0, 1, 2, ...$, of the solution, denoted by μ_ℓ, of the nonlinear function

$$\phi_\ell(\mu) = \Delta^2, \tag{4.3.4}$$

where,

$$\phi_\ell(\mu) = \mathrm{tr}(\mathcal{L}_{2\ell+1} f_\mu). \tag{4.3.5}$$

Various zero-finders can be applied, including Newton's method, cubically and quadratically convergent zero-finders ; see, e.g., [91] for discussions on these methods. Iterations of this method are designed to determine a sequence of approximations $\mu_\ell^{(j)}, j = 1, 2, 3, ...$, of μ_ℓ such that

$$\eta^2\Delta^2 \le \phi_\ell(\mu) \le \Delta^2 \tag{4.3.6}$$

To solve (4.3.4) such that (4.3.6) holds, we use the monotonically and quadratically convergent zero-finding method [56] by evaluation of the function (4.3.5) and its derivatives. Substituting the expression of $\mathcal{G}_\ell f_\mu$ given by (4.2.14) gives

$$\mathcal{G}_\ell f_\mu = E_1^T (R_{k\ell}^T R_{k\ell} + \mu I)^{-2} E_1.$$

It follows that

$$\mathcal{G}_\ell f_\mu = Z_\mu^T Z_\mu,$$

where Z_μ solves the equation

$$(R_{k\ell}^T R_{k\ell} + \mu I)Z = E_1.$$

These are the normal equations associated with the least-squares problem

$$\min_Z \left\| \begin{bmatrix} R_{k\ell} \\ \mu^{1/2} I \end{bmatrix} Z - \mu^{-1/2} \begin{bmatrix} 0 \\ E_1 \end{bmatrix} \right\|_F^2, \tag{4.3.7}$$

derivatives of $\mathrm{tr}(\mathcal{G}_\ell f_\mu)$ at μ is evaluated in the same manner as $\mathcal{G}_\ell f_\mu$. Let Z_ℓ solve (4.3.7) and introduce

$$W_\ell = (R_{k\ell}^T R_{k\ell} + \mu I)^{-1} Z_\ell,$$

which we compute by solving a least-squares problem analogous to (4.3.7). The value of the derivative is then determined by computing $-2Z_\ell^T W_\ell$. The function $\mathrm{tr}(\mathcal{L}_{2\ell+1} f_\mu)$ is evaluated analogously. In order for (4.3.3) to hold, we may have to increase ℓ further. Specifically, when (4.3.3) does not hold for $\mu = \mu_\ell^{(p)}$ that has been found by the zero-finding method iterations, we carry out one more block Lanczos step and, therefore determine approximations $\mu_{\ell+1}^{(j)}, j = 1, 2, 3, ...$, of $\mu_{\ell+1}$ of the solution, of the nonlinear function

$$\phi_{\ell+1}(\mu) = \Delta^2,$$

We assume that for some ℓ, the stopping criteria (4.3.3) is satisfied for $\mu = \mu_\ell^{(p)}$. We now want to solve the normal equation

$$(H^T H + \mu_\ell^{(p)} I)X = H^T B,$$

Let $H^T B = QR$ be the QR factorization of $H^T B$. We want to compute a sequence of approximations solutions

$$X_\ell = P_\ell^{(k)} Y_\ell, \quad \ell = 1, 2, ... \tag{4.3.8}$$

where $P_\ell^{(k)}$ is the orthonormal matrix defined in the decompositions (4.2.6) by application of ℓ steps of block Lanczos bidiagonalization algorithm to the matrix H with the initial block $P_1 = Q$. The approximate solution X_ℓ is then determined by the the following Galerkin equations

$$\begin{aligned} {P_\ell^{(k)}}^T (H^T H + \mu_\ell^{(p)} I) P_\ell^{(k)} Y_\ell &= {P_\ell^{(k)}}^T H^T B, \\ &= {P_\ell^{(k)}}^T QR, \\ &= E_1 R. \end{aligned}$$

We have

$${P_\ell^{(k)}}^T (H^T H + \mu_\ell^{(p)} I) P_\ell^{(k)} Y_\ell = (R_{k\ell}^T R_{k\ell} + \mu_\ell^{(p)} I) Y_\ell$$

We then compute the solution Y_ℓ by solving

$$(R_{k\ell}^T R_{k\ell} + \mu_\ell^{(p)} I) Y_\ell = E_1 R. \tag{4.3.9}$$

Note that (4.3.9) are the normal equations associated to the least-squares problem

$$\min_{Y_l \ell} \left\| \begin{bmatrix} R_{k\ell} \\ {\mu_\ell^{(p)}}^{1/2} I \end{bmatrix} Y_\ell - {\mu_\ell^{(p)}}^{-1/2} \begin{bmatrix} 0 \\ E_1 R \end{bmatrix} \right\|_F^2, \tag{4.3.10}$$

We use the QR factorization when solving the low-dimensional least-squares problems (4.3.7) and (4.3.10).

Algorithm 6 Outline of solution method for (4.1.9)

Input : H, B, Δ, τ, η, μ : initial guess for the zero-finding method

1. Compute $H^T B = QR$ and set $P_1 := Q \in \mathbb{R}^{n\times k}$,

2. For $\ell = 1, 2, ...$ until $E_{\ell,\mu} < \tau$

(a) Determine the matrix with orthonormal columns $P_l^{(k)}$ and the block-tridiagonal matrix $R_{k\ell}$ by the block Lanczos bidiagonalizatin when reorthogonalization is carried out

(b) Update the value μ by solving $\phi_\ell(\mu) = \Delta^2$ with the zero-finding method

3. Determine Y_ℓ by (4.3.10) and then X_ℓ by (4.3.8)

4.4 Numerical results

This section presents some representative numerical experiments. All computations were carried out using the MATLAB environment on an Intel(R) Core (TM) 2 Duo CPU T5750 computer with 3 GB of RAM. The computations were carried out with approximately 16 decimal digits of accuracy. We illustrate the efficiency of the numerical method described in this chapter by applying Algorithm 6 on a few linear discrete ill-posed problems.

Example 1

Let us consider the following Fredholm equation of the first kind in two space-dimensions

$$\int\int_{\Omega} \kappa(x, y, s, t) f(s, t) ds dt = g(x, y), \qquad (x, y) \in \Omega', \tag{4.4.1}$$

where κ is the kernel, f is the true object and g is the available data which is contaminated by measurement errors. Discretizing of the functions and approximating integration with

a quadrature rule leads to the following large-scale linear relationship

$$Kf = g, \tag{4.4.2}$$

where K is an $N \times N$ ill-conditioned matrix with $N = mn$ and some block structure depending on the imposed boundary condition, and $f, g \in \mathbb{R}^{mn}$; see, e.g, [62] for discussions on how to construct the big matrix K when the kernel is specified. Typically, the implementation of the problem (4.4.2) is a memory intensive application with formidable data. Fortunately, in many applications the kernel satisfies $\kappa(x, y; s, t) = \kappa(x - s, y - t)$, and it is said to be invariant in space or isotropic. In this case, the integration in equation (4.4.1) is a convolution operation, and thus κ can usually be represented by a compact data structure when the storage becomes an issue. Applications when the kernel is spatially invariant and therefore exploiting the matrix data structure can be found in [5, 37, 80, 82]. Solving the problem requires the use of an iterative scheme [87, 57], the most intensive part of these methods is the matrix-vector multiplications with K, which is assumed to be extremely large and therefore an efficient storage scheme exploiting structure should be used. This is the aim of the proposed method in this article. Thus, instead of applying an iterative method to the large linear system (4.4.2), we apply our Algorithm 6 to an equivalent linear system of multiple right hand side with a matrix of moderate size.

We let $F = \mathsf{mat}(f)$ and $G = \mathsf{mat}(g)$. For spatially invariant kernels, we partition F and G into cell arrays. The dimension of the cell arrays are assumed to be the same as the cell array containing the kernel. That is, for example, if F and G are partitioned into k cell arrays of size $d \times d$ each, then we transform all the cell arrays in vectors by the linear operator vec and therefore the corresponding linear system of equation of multiple right

hand side is given by

$$H * \mathsf{vec}(X_i) = \mathsf{vec}(B_i), \quad i = 1, ..., k$$

where $H \in \mathbb{R}^{d^2 \times d^2}$ is defined by the kernel in all the k cell arrays. X_i and B_i are the i-th region of F and G, respectively . We now partition F and G into appropriate size blocks based on the size of the invariant kernel κ. In the case of spatially invariant kernels, the integration in equation (4.4.3) is a convolution operation, and thus the collection of the convolution of each block with κ can be transformed into linear system with multiple right hand sides of the form (4.1.2). As a test, we consider a spatially invariant kernel whose function is given by

$$k(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}(x^2 + y^2)\right). \tag{4.4.3}$$

The test true object in our example is given by $f(x, y) = f_1(x)f_2(y)$, where

$$f_i(s) = \begin{cases} 1 + \cos(\frac{\pi}{3}s) & \text{if } |s| < 3 \\ 0 & \text{otherwise,} \end{cases} \qquad i = 1, 2.$$

We use the MATLAB code `phillips` from [61] to discretize f and obtain a scaled solution $F \in \mathbb{R}^{N \times N}$ with $N = 256$, we then partition it to $k = N/d^2$ small blocks ($d \ll N$). Let $\widehat{x}^{(i)} = \mathsf{vec}(\widehat{X}_i)$, where $\widehat{X}_i$ denote the $d \times d$ i-th block. By setting $\widehat{X} = [\widehat{x}^{(1)}, \widehat{x}^{(2)}, ..., \widehat{x}^{(k)}]$, the error-free right hand-side of (4.1.2) is given by $\widehat{B} = H\widehat{X}$, where $H \in \mathbb{R}^{d^2 \times d^2}$ is the coefficient matrix modelling the convolution operation of each block with the spatially invariant kernel (4.4.3). In this example, we choose the kernel (4.4.3) to be of size $d \times d$, where $d = 64$. By setting $\sigma = 2.5$, the MATLAB code `kronDecomp` from [61] gives a matrix $H = H_1 \otimes H_2 \in \mathbb{R}^{64^2 \times 64^2}$ of condition number equal to 1.3834×10^{21}, and where the definition of the Kronecker product of two matrices $H = (a_{ij})$ and $B = (b_{ij})$ of sizes

$n \times p$ and $s \times q$, respectively, is defined as the $(ns) \times (pq)$ matrix $H \otimes B$ with block entries $a_{ij}B$. Applications when H has a Kronecker product structure when solving Fredholm integral equations of the first kind in two space-dimensions with a separable kernel can be found in [13]. When no error E matrix is added to $\widehat{B}$, Algorithm 6 with $\eta = 0.999999$ and $\tau = 10^{-3}$ requires 2 block Lanczos bidiagonalization with a CPU time of 0.89 s and gives the approximate solution X_2, which satisfies $\frac{\|\widehat{X}-X_2\|_F}{\|\widehat{X}\|_F} = 9.99 \times 10^{-7}$. The computed value of μ is 9.89×10^{-7}. Figure 5.1 displays the true object, the spacially invariant kernel and the computed approximate solution X_2 obtained when no noise is added to the available data. We next add Gaussian white noise so that the error E has normally distributed entries with zero mean and is normalized to correspond to a the noise level $\|E\|_F/\|\widehat{B}\|_F = 0.01$. Algorithm 6 applied to this noisy system with $\eta = 0.9999$ and $\tau = 10^{-3}$ requires 2 block Lanczos bidiagonalization steps with a CPU time of 0.86 s to determine the approximate solution X_2 of relative error $\frac{\|\widehat{X}-X_2\|_F}{\|\widehat{X}\|_F} = 2.20 \times 10^{-3}$. The computed value of μ is 1.49×10^{-4}. Figure 5.1 displays the computed approximate solution X_2 obtained when a 0.01 noise level is added to the available data. We compare our approach with several existing methods which are closely related. The first related method proposed in [25] utilizes the connection between (standard) Golub-Kahan bidiagonalization and Gauss quadrature rules for solving large ill-conditioned linear systems of equations (4.4.4). We refer to this method as GKB. Another recent related method is GLT (Algorithm 2). This method use the relation between the global Lanczos method and Gauss-type quadrature rules to solve the problem (4.4.4) when K is given as a Kronecker product for the spatially invariant case. In Table 4.1, we report more results for comparing the three methods for two levels of noise. We choose $\eta = 0.9999$ for the three methods. Looking at the results

displayed in Table 4.1 we can state that, for this example, Algorithm 6 exhibits excellent performance both in terms of quality of the results and computational time.

TABLE 4.1 – Results for Example 1.

Noise level	Method	μ	R(X)	CPU-times(s)	l
10^{-3}	Algorithm 6	2.49×10^{-4}	2.49×10^{-4}	0.89	2
	GLT	1.19×10^{-4}	7.50×10^{-2}	7.31	61
	GKB	6.09×10^{-5}	1.93×10^{-2}	33.46	131
10^{-2}	Algorithm 6	3.96×10^{-4}	2.30×10^{-3}	0.87	2
	GLT	8.04×10^{-4}	3.82×10^{-2}	5.77	51
	GKB	7.55×10^{-4}	4.44×10^{-2}	21.18	85

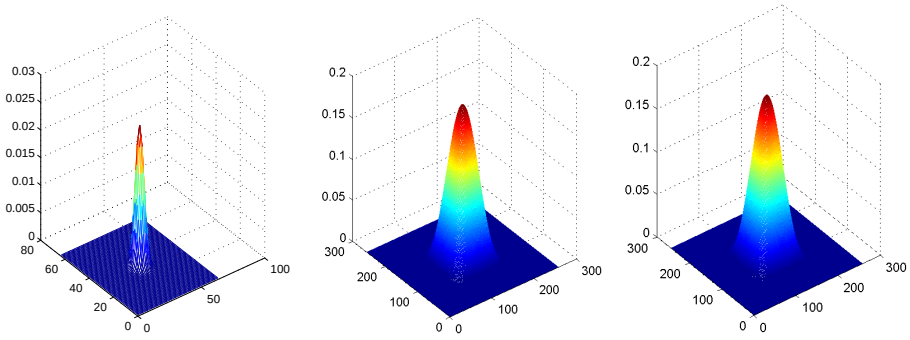


FIGURE 4.1 – Example 1 : Spatially invariant kernel (left), true object (center), and approximate solution (right) when no noise is added to the data.

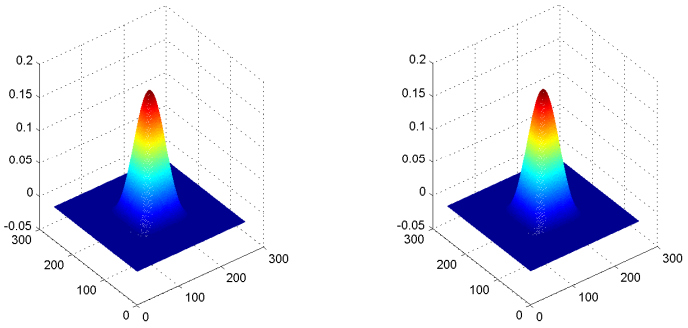


FIGURE 4.2 – Example 1 : Approximate solutions for 0.001 (left) and for 0.01 noise level (right).

Example 2

As a second test, we consider the spatially invariant kernel whose function is given by (4.4.3). The test true object in our example is given by $g(x, y) = g_1(x)g_1(y)$, where

$$g_i(s) = 2\sinh(s)/s, \quad i = 1, 2.$$

We use the MATLAB code `baart` from [61] to discretize g and obtain a scaled solution $F \in \mathbb{R}^{N\times N}$ with $N = 256$, we then partition it to 16 small blocks of sizes 64×64. We construct $\widehat{X}$ and the error-free right hand-side $\widehat{B}$, by the same way as in Example 2. Table 4.2 displays the computed regularization parameters and the relative error in computed approximate solutions determined by Algorithm 6 with $\eta = 0.98$ and $\tau = 10^{-3}$ for different noise levels, as well as the number of block Lanczos steps (BLS) required to satisfy the stopping criterion.

TABLE 4.2 – Results for Example 3.

Noise level	BLS	Regularization parameter	Relative error
0.01	2	2.04×10^{-2}	2.17×10^{-2}
0.1	7	2.21×10^{-2}	7.72×10^{-2}

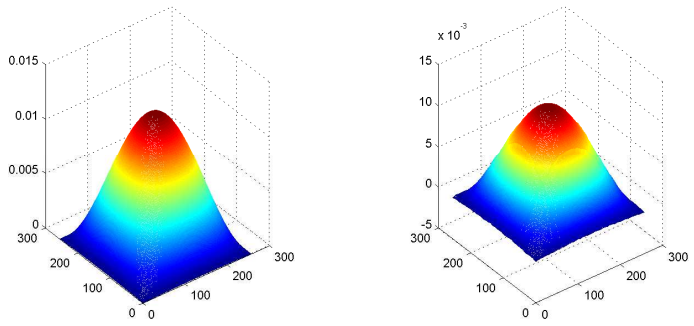


FIGURE 4.3 – Example 2 : True object (left) and approximate solution (right).

Example 3

In this example, we describe an image restoration problem by sectioning the image and applying the block restoration scheme to illustrate the main advantage of the approach discussed in this chapter. An efficient way to reduce memory use is to process the image in blocks. We will see how block processing can produce better results in terms of accuracy and execution time by comparing our approach with several existing methods that are closely related to ours. In order to understand this numerical experiment, we first recall that the model of image restoration is expressed as

$$Kx = g, \quad g = \widehat{g} + e, \tag{4.4.4}$$

where g is the blurred and noisy image, $\widehat{g}$ is the associated blurred and noise-free image. K is a blurring matrix and and it models the blurring operation, x is the true image and e is the additive noise. The blurring matrix K is determined from two ingredients : the point spread function (PSF) [62], which defines how each pixel is blurred, and the boundary conditions, which specify our assumptions on the scene just outside our image. We describe the strategy related to our approach to solve the problem (4.4.4), when a spatially invariant point spread function is given. That is, the product Kx is usually considered as a convolution of the associated PSF and the object x. We assume now that the overall space invarying PSF does not vary across the image, and thus the PSF can be obtained from any region of the image. Taking into consideration this important property, it is necessary to exploit the block processing of the image. This discussion leads us to decompose the blurred and noisy image into sets of patches on which the blur is assumed to be the same. Given the blurred and noisy image partitioned into small blocks based on the size of the point spread function, we then assume that every blurred and noisy subimage is given by convolving the PSF with the exact subimage. Collection of image restoration problems for each block image will be transformed into a linear system with multiple right hand sides. In many cases, when processing an image by blocks it is necessary to overlap the regions in order to minimize edge effects. We now want to partition the problem (4.4.4) as

$$Hx_i = g_i, \quad i = 1, ..., k,$$

where H is a matrix of moderate size based on the size of the PSF, x_i and g_i are vectors obtained by transforming each subimage block of the true image and the blurred and noisy

image, respectively, by the linear operator vec. k is the overall number of subimages. We assume that the original image has dimension $n \times n$, therefore K is an $N \times N$ matrix, where $N = n^2$. Let k be the number of divided subimages X_i of equal size $d \times d$. The corresponding blurred subimages $\widehat{g}_i$ are obtained by convolving each subimage X_i with the PSF. By setting $x_i = \mathsf{vec}(X_i)$, $\widehat{g}_i = \mathsf{vec}(\widehat{g}_i)$, $X = [x_1, ..., x_k]$ and $\widehat{G} = [\widehat{g}_1, ..., \widehat{g}_k]$, the image restoration problem (4.4.4) is restated by the following form with k right-hand sides

$$HX = G, \quad G = \widehat{G} + E \tag{4.4.5}$$

where H is the $d^2 \times d^2$ matrix corresponding to the PSF and E is the measurement noise. At this level, we are now able to us our approach described by Algorithm 6 to solve the problem (4.4.5). The test image used in our study is the Airplane image of size 256×256 pixels, shown on the left of Figure 4.4. We then consider a PSF array for out-of-focus blur [61], where the entries are given by

$$(\text{PSF})_{ij} = \begin{cases} \frac{1}{(\pi r^2)}, & \text{if} \quad (i-k)^2 + (j-l)^2 \leq r^2, \\ 0 & \text{else} \end{cases}$$

where (k, l) is the center of PSF, and r is the radius of the blur. As a test, we set $r = 5$. We now want to corrupt the available image by the spatially invariant blur given by the above PSF. The test image is divided into 64 subimages of small size, where each subimage is of size 32×32 pixels. We convolve each subimage with the PSF and we add a Gaussian white noise with 10^{-3} noise level to obtain the blurred and noisy image represented in the right of Figure 4.4. In the case of spatially invariant blur, the blurring matrix H given in (4.4.5) can be decomposed as a Kronecker product $H = H_1 \otimes H_2$. It is well known that many blurring matrices have Kronecker structure or can be approximated well by a

matrix with this structure ; see [71, 72, 99]. We use the MATLAB code kronDecomp from [61] to obtain the two matrices, H_1 and H_2 of size 32×32. To minimize edge effect, we use the reflexive boundary conditions. The performances of the method proposed in this chapter (Algorithm 6) and the GKB and GLT methods are compared by computing the peak signal to noise ratio (PSNR) defined by

$$\mathrm{PSNR}(X) = 20\log_{10}\left(\frac{255}{\|\widehat{X} - X\|_F}\right),$$

where 255 is the maximum possible pixel value of the image. We also compute the following relative error to evaluate the precision of the estimates

$$R(X_l) = \frac{\|\widehat{X} - X_l\|_F}{\|\widehat{X}\|_F},$$

where $\widehat{X}$ and X_ℓ are the true and restored images, respectively. Using Algorithm 6 with $\eta = 0.9994$ and $\tau = 10^{-3}$ gives the restored image obtained represented on the right of Figure 4.7. The relative error was $R(X_\ell) = 2.62 \times 10^{-2}$ with the $\mathrm{PSNR}(X_\ell) = 33.88$. The Algorithm 6 was terminated after 5 iterations of block Lanzos bidiagonalization (BLB) with a CPU time of 1.54 s. The computed optimal value of μ is 2.80×10^{-4}. Table 4.3 reports on more results for comparing the quality of the restorations obtained by the three methods used to solve the present problem in terms of the PSNR and also in the execution time and relative error for three levels of noise. We observe that Algorithm 6 can deliver better results than the other ones. In Figure 4.5 we plot the values of the relative error and the regularization parameter versus the number of iterations, obtained by applying Algorithm 6 when the noise level in the data is 10^{-3}.

TABLE 4.3 – Results for Example 3.

Noise level	Method	μ	PSNR	$R(X_l)$	CPU-times(s)	ℓ
10^{-4}	Algorithm 6	1.86×10^{-4}	36.61	1.91×10^{-2}	1.74	5
	GLT	1.34×10^{-4}	34.86	2.34×10^{-2}	3.52	72
	GKB	1.65×10^{-4}	35.92	2.07×10^{-2}	16.97	67
10^{-3}	Algorithm 6	2.80×10^{-4}	33.88	2.62×10^{-2}	1.54	5
	GLT	1.95×10^{-4}	33.49	2.74×10^{-2}	3.63	68
	GKB	2.65×10^{-4}	33.24	2.82×10^{-2}	15.61	62
10^{-2}	Algorithm 6	8.80×10^{-3}	28.50	4.86×10^{-2}	0.22	3
	GLT	6.80×10^{-3}	27.57	5.41×10^{-2}	0.31	6
	GKB	7.90×10^{-3}	27.47	5.74×10^{-2}	2.19	6

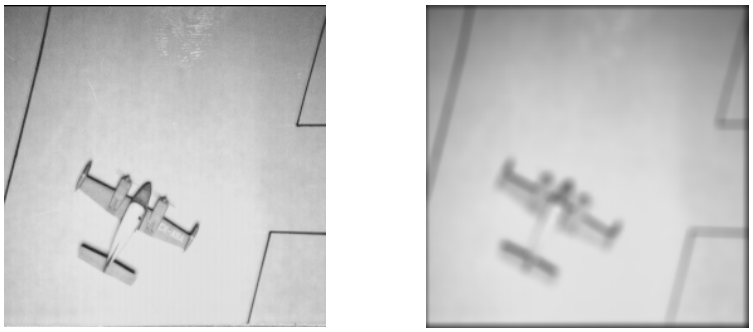


FIGURE 4.4 – Example 3 : Original image (left), blurred and noisy image (right).

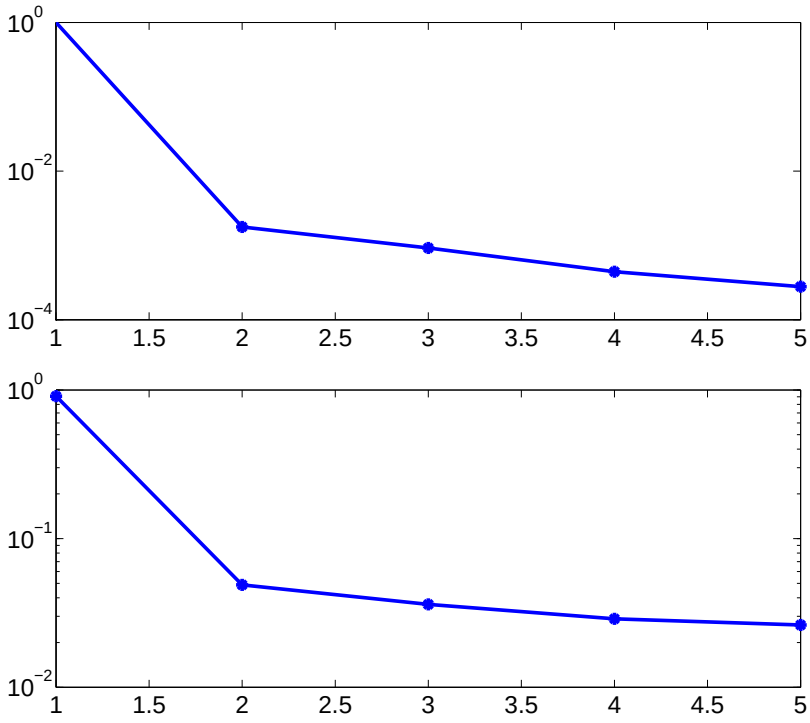


FIGURE 4.5 – Example 3 : Values of the relative error (lower graph) and the regularization parameter (upper graph) versus the number of iterations, with noise level 10^{-3}.

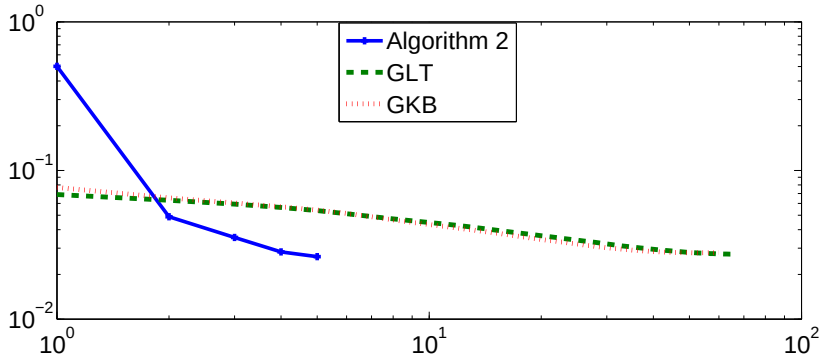


FIGURE 4.6 – Example 3 : Behaviour of the relative error versus the number of iterations for different methods with noise level 10^{-3}.

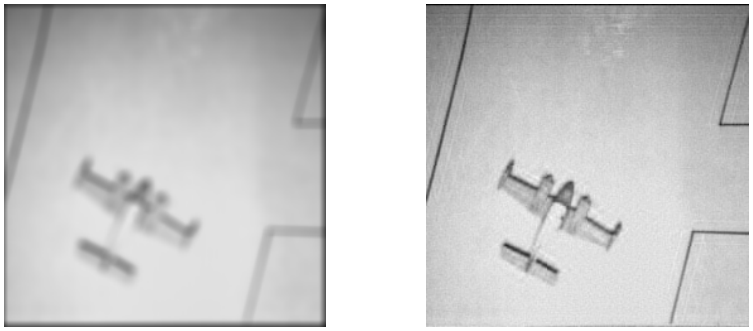


FIGURE 4.7 – Example 3 : Blurred and noisy image (left), restored image (right)

Chapitre 5

Solution methods for linear discrete ill-posed problems for color image restoration

5.1 Introduction

This chapter discusses the use of iterative methods based on standard or block Golub–Kahan-type bidiagonlization, combined with Tikhonov regularization, to the restoration of a multi-channel image from an available blur- and noise-contaminated version. Applications include the restoration of color images whose RGB (red, green, and blue) representation uses three channels ; see [44, 62]. The methods described also can be applied to the solution of Fredholm integral equations of the first kind in two or more space dimensions and to the restoration of hyper-spectral images. The latter kind of images generalize color images in that they allow more than three "colors" ; see, e.g., [77]. For definiteness, we focus in this section on the restoration of k-channel images that have been contaminated by blur and noise, and formulate this restoration task as a linear system of equations with k right-hand side vectors, where each spectral band corresponds to one channel. To

simplify our notation we assume the image to be represented by an array of $n \times n$ pixels in each one of the k channels, where $1 \leq k \ll n^2$. Let $b^{(i)} \in \mathbb{R}^{n^2}$ represent the available blur- and noise- contaminated image in channel i, let $e^{(i)} \in \mathbb{R}^{n^2}$ describe the noise in this channel, and let $\widehat{x}^{(i)} \in \mathbb{R}^{n^2}$ denote the desired unknown blur- and noise-free image in channel i. The corresponding quantities for all k channels $b, \widehat{x}, e \in \mathbb{R}^{n^2 k}$ are obtained by stacking the vectors $b^{(i)}, \widehat{x}^{(i)}, e^{(i)}$ of each channel. For instance, $b = [(b^{(1)})^T, \ldots, (b^{(k)})^T]^T$. The degradation model is of the form

$$b = H\widehat{x} + e \tag{5.1.1}$$

with blurring matrix

$$H = A_k \otimes A \in \mathbb{R}^{n^2 k \times n^2 k}.$$

The matrix $A \in \mathbb{R}^{n^2 \times n^2}$ represents within-channel blurring, which is assumed to be the same in all channels, and the small matrix $A_k \in \mathbb{R}^{k \times k}$ models cross-channel blurring. When $A_k = I_k$, there is no cross-channel blurring. In this situation, the blurring is said to be within-channel only and the deblurring problem decouples into k independent deblurring problems. The degradation model (5.1.1) then can be expressed in the form

$$B = A\widehat{X} + E, \tag{5.1.2}$$

where $B = [b^{(1)}, \ldots, b^{(k)}]$, $\widehat{X} = [\widehat{x}^{(1)}, \ldots, \widehat{x}^{(k)}]$, and $E = [e^{(1)}, \ldots, e^{(k)}]$ are $n^2 \times k$ "block vectors".

When $A_k \neq I_k$, the degradation model (5.1.1) can be expressed as

$$B = \mathcal{A}(\widehat{X}) + E, \tag{5.1.3}$$

where the matrices B, $\widehat{X}$, and E are the same as in (6.1.9) and the linear operator $\mathcal{A}$ is defined by

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{n^2 \times k} &\rightarrow \mathbb{R}^{n^2 \times k} \\ \mathcal{A}(X) &:= AXA_k^T. \end{aligned} \tag{5.1.4}$$

Its transpose is given by $\mathcal{A}^T(X) := A^T X A_k$. For notational simplicity, we denote in the following both the matrix A in (6.1.9) and the linear operator $\mathcal{A}$ in (6.1.10) by A, and we write $\mathcal{A}(X)$ as AX.

The singular values of a blurring A matrix or operator typically "cluster" at the origin, i.e., A has many singular values of different orders of magnitude close to zero. It follows that the solution (if it exists) of the linear system of equations

$$AX = B \tag{5.1.5}$$

is very sensitive to the error E in B. Linear systems of equations with a matrix of this kind are commonly referred to as linear discrete ill-posed problems.

Let $\widehat{B}$ denote the (unknown) noise-free block vector associated with B. The system of equations $AX = \widehat{B}$ is assumed to be consistent. Denote the solution of minimal Frobenius norm by $\widehat{X}$. We would like to determine an accurate approximation of $\widehat{X}$ given B and A. This generally is a difficult computational task due to the error E in B and the presence of tiny positive singular values of A.

Tikhonov regularization reduces the sensitivity of the solution of (5.1.5) to the error E in B by replacing (5.1.5) by a penalized least-squares problem of the form

$$\min_{X \in \mathbb{R}^{n^2 \times k}} \{\|AX - B\|_F^2 + \mu^{-1}\|X\|_F^2\}, \tag{5.1.6}$$

where $\mu > 0$ is the regularization parameter. The normal equations associated with this minimization problem are given by

$$(A^T A + \mu^{-1} I)X = A^T B. \tag{5.1.7}$$

They have the unique solution

$$X_\mu = \left(A^T A + \mu^{-1} I\right)^{-1} A^T B \tag{5.1.8}$$

for any $\mu > 0$. The size of μ determines how sensitive X_μ is to the error in B and how close X_μ is to the desired solution $\widehat{X}$. We will comment on the use of μ^{-1} in (5.1.6) instead of μ below.

The computation of an accurate approximation X_μ of $\widehat{X}$ requires that a suitable value of the regularization parameter μ be used. Several methods for determining such a μ-value have been suggested in the literature. These include so-called heuristic methods that do not require any knowledge of the size of the error E in B, such as the L-curve criterion, generalized cross validation, and the quasi-optimality criterion ; see, e.g., [22, 36, 42, 58, 69, 70, 90] for discussions and illustrations. We will use the discrepancy principle, discussed, e.g., in [31], to determine μ in the computed examples reported in Section 5.5. The discrepancy principle requires that a bound $\epsilon > 0$ of $\|E\|_F$ be available and prescribes that $\mu > 0$ be chosen so that the solution (5.1.8) of (5.1.6) satisfies

$$\|B - AX_\mu\|_F = \eta\epsilon, \tag{5.1.9}$$

where $\eta > 1$ is a user-specified constant independent of ϵ. A zero-finder can be applied to determine a μ-value such that the associated Tikhonov solution (5.1.8) satisfies (5.1.9).

When the matrix A is of small to moderate size, the left-hand side of (5.1.9) easily can be evaluated by using the singular value decomposition (SVD) of A. However, computation

of the SVD is impractical when the matrix A is large. We will discuss how an approximate solution of (5.1.6) can be computed by first evaluating a partial block Golub–Kahan bidiagonalization (BGKB) of A and then solving (5.1.6) in a subspace so defined.

Alternatively, we may reduce A to a small bidiagonal matrix with the aid of global Golub–Kahan bidiagonalization (GGKB), which also is a block method, and then apply the connection between GGKB and Gauss-type quadrature rules to determine upper and lower bounds for the left-hand side of (5.1.9). This allows the computation of a suitable value of μ in a simple manner. This approach has previously been applied in [10] ; the GGKB method was first described in [95]. The BGKB and GGBK block methods are compared to the application of Golub–Kahan bidiagonalization (with block size one) in two ways : One approach applies Golub–Kahan bidiagonalization with initial vector $b^{(1)}$ and generates a solution subspace that is large enough to solve all systems of equations

$$Ax^{(i)} = b^{(i)}, \qquad i = 1, \ldots, k, \tag{5.1.10}$$

with Tikhonov regularization. The other approach is to simply solve each one of the k systems of equations (5.1.10) independently with Golub–Kahan bidiagonalization and Tikhonov regularization, i.e., by using the algorithm described in [25] k times.

This chapter is organized as follows. Section 5.2 describes the BGKB method and discusses its application to the solution of (5.1.6). We remark that the bidiagonalization method differs from the one described by Golub et al. [45]. The determination of a regularization parameter such that the computed solution satisfies the discrepancy principle is also discussed. Section 6.4 reviews the use of the GGBK method to reduce A. The connection between this reduction and Gauss-type quadrature rules is exploited to

compute bounds for the left-hand side of (5.1.9). The solution of (5.1.6) by applying Golub–Kahan bidiagonalization (with block size one) determined by A and the initial vector $b^{(1)}$ is described in Section 5.4. Sufficiently many bidiagonalization steps are carried out so that all systems (5.1.10) can be solved with solution subspaces determined by A and $b^{(1)}$. We also discuss the solution of the k systems (5.1.10) independently by Golub–Kahan bidiagonalization and Tikhonov regularization as described in [25]. Section 5.5 presents a few numerical examples.

5.2 Solution by partial block Golub–Kahan bidiagonalization

Introduce for $\mu > 0$ the function

$$\phi(\mu) = \|B - AX_\mu\|_F^2. \tag{5.2.1}$$

Substituting (5.1.8) into (5.2.1) and using the identity

$$I - A(A^TA + \mu^{-1}I)^{-1}A^T = (\mu AA^T + I)^{-1} \tag{5.2.2}$$

shows that (5.2.1) can be written as

$$\phi(\mu) = \mathrm{tr}\left(B^T f_\mu(AA^T)B\right) \tag{5.2.3}$$

with

$$f_\mu(t) = (\mu t + 1)^{-2}. \tag{5.2.4}$$

The determination of a value of the regular parameter $\mu > 0$ that satisfies (5.1.9) generally requires the function ϕ to be evaluated for several μ-values. Each evaluation

of ϕ is very expensive for large-scale problems. We therefore approximate the expression $B^T f_\mu(AA^T)B$ by a simpler one, which we determine with a few steps of block Golub–Kahan bidiagonalization as follows. Introduce the QR factorization $B = P_1 R_1$, where $P_1 \in \mathbb{R}^{n^2 \times k}$ has orthonormal columns and $R_1 \in \mathbb{R}^{k \times k}$ is upper triangular. Then ℓ steps of the BGKB method applied to A with initial block vector P_1 gives the decompositions

$$AQ_\ell^{(k)} = P_{\ell+1}^{(k)} \bar{C}_\ell^{(k)}, \qquad A^T P_\ell^{(k)} = Q_\ell^{(k)} C_\ell^{(k)^T}, \tag{5.2.5}$$

where the matrices $P_{\ell+1}^{(k)} = [P_1, \ldots, P_{\ell+1}] \in \mathbb{R}^{n^2 \times (\ell+1)k}$ and $Q_\ell^{(k)} = [Q_1, \ldots, Q_\ell] \in \mathbb{R}^{n^2 \times \ell k}$ have orthonormal columns and

$$\bar{C}_\ell^{(k)} := \begin{bmatrix} L_1 & & & & \\ R_2 & L_2 & & & \\ & \ddots & \ddots & & \\ & & & R_\ell & L_\ell \\ & & & & R_{\ell+1} \end{bmatrix} \in \mathbb{R}^{k(\ell+1) \times k\ell}, \tag{5.2.6}$$

is lower block bidiagonal with lower triangular diagonal blocks $L_j \in \mathbb{R}^{k \times k}$ and upper triangular blocks $R_j \in \mathbb{R}^{k \times k}$. Moreover, $C_\ell^{(k)}$ is the leading $k\ell \times k\ell$ submatrix of $\bar{C}_\ell^{(k)}$ and P_ℓ is the leading $n^2 \times \ell k$ submatrix of $P_{\ell+1}$. For block size $k = 1$, the decompositions (5.2.5) simplify to the decompositions computed by the algorithm bidiag1 by Paige and Saunders [87]. In particular, the decompositions (5.2.5) differ from the ones described by Golub et al. [45], who compute an upper block bidiagonal matrix. In our discussion, we will assume that ℓ is small enough so that the triangular matrices L_j, $j = 1, \ldots, \ell$, and R_j, $j = 2, \ldots, \ell + 1$, are nonsingular.

It follows from (5.2.5) that the range of the matrix $P_\ell^{(k)}$ is the block Krylov subspace

$$\mathbb{K}_\ell(AA^T, B) = \text{Range}[P_1, AA^T P_1, (AA^T)^2 P_1, \ldots, (AA^T)^{\ell-1} P_1].$$

Similarly, the range of the matrix $Q_\ell^{(k)}$ is the block Krylov subspace

$$\mathbb{K}_\ell(A^T A, A^T B) = \text{Range}[A^T P_1, A^T AA^T P_1, (A^T A)^2 A^T P_1, \ldots, (A^T A)^{\ell-1} A^T P_1].$$

Multiplying the rightmost equation (5.2.5) by A from the left yields

$$AA^T P_\ell^{(k)} = P_{\ell+1}^{(k)} \bar{C}_\ell^{(k)} C_\ell^{(k)^T}. \tag{5.2.7}$$

Therefore,

$$P_\ell^{(k)^T} AA^T P_\ell^{(k)} = C_\ell^{(k)} C_\ell^{(k)^T}.$$

This suggests that $f_\mu(AA^T)$ may be approximated by $f_\mu(C_\ell^{(k)} C_\ell^{(k)^T})$, which is much easier to evaluate than $f_\mu(AA^T)$. It follows from results by Golub and Meurant [46, 47] on the symmetric block Lanczos algorithm that the expression

$$\mathcal{G}_\ell f_\mu = R_1^T E_1^T f_\mu \left(C_\ell^{(k)} C_\ell^{(k)^T} \right) E_1 R_1 \tag{5.2.8}$$

can be interpreted as an ℓ-block Gauss quadrature rule for the approximation of $B^T f_\mu(AA^T)B$, i.e.,

$$\mathcal{G}_\ell f = B^T f(AA^T)B \qquad \forall f \in \mathbb{P}_{2\ell-1},$$

where $\mathbb{P}_{2\ell-1}$ denotes the set of all polynomials of degree at most $2\ell - 1$; see also [34] for related discussions. We therefore approximate (5.2.3) by

$$\phi_\ell(\mu) = \text{tr}(\mathcal{G}_\ell f_\mu) \tag{5.2.9}$$

and let the regularization parameter be the solution of

$$\phi_\ell(\mu) = \eta^2 \epsilon^2. \tag{5.2.10}$$

The following result shows that $\phi_\ell(\mu)$ is decreasing and convex. This makes it convenient to compute the solution μ_ℓ of (5.2.10) by Newton's method ; see below.

Proposition 5.2.1. *The functions $\phi(\mu)$ and $\phi_\ell(\mu)$, defined by* (5.2.3) *and* (5.2.9) *for $\mu > 0$, respectively, satisfy*

$$\phi'(\mu) < 0, \qquad \phi''(\mu) > 0, \qquad \phi'_\ell(\mu) < 0, \qquad \phi''_\ell(\mu) > 0.$$

Démonstration. The derivative of $\phi(\mu)$ is given by

$$\phi'(\mu) = -2\mathrm{tr}(B^T(\mu AA^T + I)^{-3}AA^TB).$$

It follows from $(\mu AA^T + I)^{-1}A = A(\mu A^TA + I)^{-1}$ that

$$\phi'(\mu) = -2\mathrm{tr}(B^TA(\mu A^TA + I)^{-3}A^TB).$$

Substituting the spectral factorization $A^TA = S\Lambda S^T$, $S^TS = I$, into the above expression and letting $W = [w_1, \ldots, w_k] = S^TA^TB$ yields

$$\phi'(\mu) = -2\mathrm{tr}(W^T(\mu\Lambda + I)^{-3}W) = -2\sum_{j=1}^{k} w_j^T(\mu\Lambda + I)^{-3}w_j < 0.$$

Thus, $\phi(\mu)$ is a decreasing function of μ. Turning to the second derivative, we have

$$\phi''(\mu) = 6\mathrm{tr}(B^TAA^T(\mu AA^T + I)^{-4}AA^TB),$$

and can proceed similarly as above to show that $\phi''(\mu) > 0$.

The derivative of $\phi_\ell(\mu)$ is given by

$$\phi'_\ell(\mu) = \mathrm{tr}(R_1^TE_1^TC_\ell^{(k)}(\mu C_\ell^{(k)^T}C_\ell^{(k)} + I)^{-3}C_\ell^{(k)^T}E_1R_1), \tag{5.2.11}$$

where we again use the identity $(\mu C_\ell^{(k)} C_\ell^{(k)^T} + I)^{-1} C_\ell^{(k)} = C_\ell^{(k)} (\mu C_\ell^{(k)^T} C_\ell^{(k)} + I)^{-1}$. The stated properties of $\phi'_\ell(\mu)$ and $\phi''_\ell(\mu)$ can be shown by substituting the spectral factorization of $C_\ell^{(k)^T} C_\ell^{(k)}$ into (5.2.11). □

Since $\phi_\ell(\mu)$ is decreasing and convex, Newton's method converges monotonically and quadratically to the solution μ_ℓ of (5.2.10) for any initial approximate solution $\mu_{\text{init}} < \mu_\ell$. This makes it easy to implement the Newton method. For instance, we may use $\mu_{\text{init}} = 0$ when ϕ_ℓ and its derivative are suitably defined at $\mu = 0$; see [25] for a detailed discussion of the case when the block size is one.

We note that the function $\mu \to \phi_\ell(1/\mu)$, which corresponds to the regularization term $\mu \|X\|_F^2$ in (5.1.6), is not guaranteed to be convex. Therefore, Newton's method has to be safeguarded when applied to the solution of $\phi_\ell(1/\mu) = \epsilon^2$. This is the reason for considering Tikhonov regularization of the form (5.1.6).

Proposition 5.2.2. *Let $P_{\mathcal{N}(M)}$ denote the orthogonal projector onto the null space $\mathcal{N}(M)$ of the matrix M. Then*

$$\phi(0) = \operatorname{tr}(B^T B), \qquad \lim_{\mu \to \infty} \phi(\mu) = \operatorname{tr}(B^T P_{\mathcal{N}(AA^T)} B),$$

$$\phi_\ell(0) = \operatorname{tr}(B^T B), \qquad \lim_{\mu \to \infty} \phi_\ell(\mu) = \operatorname{tr}(R_1^T E_1^T P_{\mathcal{N}(R_\ell R_\ell^T)} E_1 R_1).$$

Démonstration. The value at zero and limit of ϕ follow from (5.2.3). The expression (5.2.8) and the definition of the upper triangular matrix R_1 yield

$$\phi_\ell(0) = \operatorname{tr}\left(R_1^T E_1^T f_0\left(C_\ell^{(k)} C_\ell^{(k)^T} \right) E_1 R_1 \right) = \operatorname{tr}(R_1^T R_1) = \operatorname{tr}(B^T B).$$

The result for $\phi_\ell(\mu)$ as $\mu \to \infty$ follows similarly as for ϕ. □

Let the regularization parameter μ_ℓ be computed by Newton's method. We then determine the corresponding approximate solution by projecting the normal equations (5.1.7) with $\mu = \mu_\ell$ onto a smaller space determined by the decompositions (5.2.5). We seek to determine an approximate solution of the form

$$X_{\mu_\ell} = Q_\ell^{(k)} Y_{\mu_\ell}, \qquad Y_{\mu_\ell} \in \mathbb{R}^{k\ell \times k\ell}, \tag{5.2.12}$$

by solving the normal equations by a Galerkin method,

$$(Q_\ell^{(k)})^T (A^T A + \mu_\ell^{-1} I) Q_\ell^{(k)} Y_{\mu_\ell} = (Q_\ell^{(k)})^T A^T B, \tag{5.2.13}$$

which simplifies to

$$(\bar{C}_\ell^{(k)^T} \bar{C}_\ell^{(k)} + \mu_\ell^{-1} I) Y_{\mu_\ell} = \bar{C}_\ell^{(k)^T} E_1 R_1. \tag{5.2.14}$$

We compute the solution $Y_{\ell,\mu}$ by solving a least-squares problem for which (5.2.14) are the normal equations

$$\min_{Y \in \mathbb{R}^{k\ell \times k\ell}} \left\| \begin{bmatrix} \bar{C}_\ell^{(k)} \\ \mu_\ell^{-1/2} I \end{bmatrix} Y - \mu_\ell^{1/2} \begin{bmatrix} E_1 R_1 \\ 0 \end{bmatrix} \right\|_F^2. \tag{5.2.15}$$

Our reason for computing the solution of (5.2.15) instead of (5.2.14) is that solving the least-squares problem is less sensitive to errors for small values of $\mu_\ell > 0$.

Proposition 5.2.3. *Let μ_ℓ solve* (5.2.10)*, and let Y_{μ_ℓ} solve* (5.2.13)*. Then the associated approximate solution $X_{\mu_\ell} = Q_\ell^{(k)} Y_{\mu_\ell}$ of* (5.1.6) *satisfies*

$$\|AX_{\mu_\ell} - B\|_F^2 = tr\left(R_1^T E_1^T f_{\mu_\ell}(\bar{C}_\ell^{(k)} \bar{C}_\ell^{(k)^T}) E_1 R_1 \right).$$

Démonstration. Using the expression of $X_{\ell,\mu}$ and applying (5.2.5) shows that

$$\begin{aligned} AX_{\mu_\ell} - B &= AQ_\ell^{(k)} Y_{\mu_\ell} - B \\ &= P_{\ell+1}^{(k)} \bar{C}_\ell^{(k)} Y_{\mu_\ell} - P_1 R_1 \\ &= P_{\ell+1}^{(k)} \left(\bar{C}_\ell^{(k)} Y_{\mu_\ell} - E_1 R_1 \right), \end{aligned}$$

where we recall that $B = P_1 R_1$. It follows from (5.2.14) that

$$P_{\ell+1}^{(k)} \left(\bar{C}_\ell^{(k)} Y_{\mu_\ell} - E_1 R_1 \right) = P_{\ell+1}^{(k)} \left[\left(\bar{C}_\ell^{(k)} \left(\bar{C}_\ell^{(k)^T} \bar{C}_\ell^{(k)} + \mu_\ell^{-1} I \right)^{-1} \bar{C}_\ell^{(k)^T} - I \right) E_1 R_1 \right].$$

The identity (5.2.2) with A replaced by $\bar{C}_\ell^{(k)}$ now yields

$$\|AX_{\mu_\ell} - B\|_F^2 = \operatorname{tr} \left(R_1^T E_1^T f_{\mu_\ell}(\bar{C}_\ell^{(k)} \bar{C}_\ell^{(k)^T}) E_1 R_1 \right).$$

□

Algorithm 7 The BGKB-Tikhonov method

Input : A, B, k, ϵ, $\eta \geq 1$.

1. Compute the QR factorization $B = P_1 R_1$.

2. For $\ell = 1, 2, \ldots$ until $\|AX_{\mu_\ell} - B\|_F \leq \eta\epsilon$

(a) Determine $Q_\ell^{(k)}$ and $P_{\ell+1}^{(k)}$ and block bidiagonal matrix $C_\ell^{(k)}$ by BGKB.

(b) Update the value μ_ℓ by solving (5.2.10) with Newton's method.

3. Determine Y_{μ_ℓ} by solving (5.2.15) and then X_{μ_ℓ} by (5.2.12).

5.3 The GGKB method and Gauss-type quadrature

We discuss the application of the GGKB method to compute an approximate solution of (5.1.6) and review how the method can be used to compute inexpensive upper and lower

bounds for the discrepancy (5.1.9). These bounds help us to determine the regularization parameter. This approach of solving (5.1.6) and determining bounds for the discrepancy has recently been described in [10], where further details can be found.

Application of ℓ steps of the GGKB method to A with initial block vector B determines the lower bidiagonal matrix

$$\bar{C}_\ell = \begin{bmatrix} \rho_1 & & & & & \\ \sigma_2 & \rho_2 & & & & \\ & \ddots & \ddots & & & \\ & & \sigma_{\ell-1} & \rho_{\ell-1} & & \\ & & & \sigma_\ell & \rho_\ell & \\ & & & & \sigma_{\ell+1} & \end{bmatrix} \in \mathbb{R}^{(\ell+1)\times\ell}$$

as well as the matrices

$$U_{\ell+1}^{(k)} = [U_1, U_2, \ldots, U_{\ell+1}] \in \mathbb{R}^{n^2\times(\ell+1)k}, \quad V_\ell^{(k)} = [V_1, V_2, \ldots, V_\ell] \in \mathbb{R}^{n^2\times\ell k}$$

with block columns $U_i, V_j \in \mathbb{R}^{n^2\times k}$, where $U_1 = s_1 B$ and $s_1 > 0$ is a scaling factor. Introduce the inner product

$$\langle F, G\rangle = \mathrm{tr}(F^T G), \qquad F, G \in \mathbb{R}^{n^2\times k}.$$

We have $\|F\|_F = \langle F, F\rangle^{1/2}$. The block columns $U_1, \ldots, U_{\ell+1}$ are orthonormal with respect to this inner product, and so are the block columns $V_1, \ldots, V_\ell$. Thus,

$$\langle U_i, U_j\rangle = \langle V_i, V_j\rangle = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$

We assume that ℓ is small enough so that all nontrivial entries of the matrix $\bar{C}_\ell$ are positive. This is the generic situation. We denote the leading $\ell\times\ell$ submatrix of $\bar{C}_\ell$ by C_ℓ.

The matrices determined satisfy

$$\big[A(V_1), A(V_2), \ldots, A(V_\ell)\big] = U_{\ell+1}^{(k)}(\bar{C}_\ell \otimes I_k), \tag{5.3.1}$$

$$\big[A^T(U_1), A^T(U_2), \ldots, A^T(U_\ell)\big] = V_\ell^{(k)}(C_\ell^T \otimes I_k). \tag{5.3.2}$$

Consider the functions (of μ),

$$\mathcal{G}_\ell f_\mu = \|B\|_F^2 e_1^T(\mu C_\ell C_\ell^T + I_\ell)^{-2} e_1, \tag{5.3.3}$$

$$\mathcal{R}_{\ell+1} f_\mu = \|B\|_F^2 e_1^T(\mu \bar{C}_\ell \bar{C}_\ell^T + I_{\ell+1})^{-2} e_1. \tag{5.3.4}$$

The function (5.3.3) can be interpreted as an ℓ-point Gauss quadrature rule for the approximation of the expression $\phi(\mu)$ defined by (5.2.3); see, e.g., [10, 36]. Similarly, the function (5.3.4) may be regarded as an $(\ell+1)$-point Gauss–Radau quadrature rule with a fixed node at the origin for the approximation of (5.2.3). The remainder formulas for Gauss and Gauss–Radau quadrature, together with the observations that the derivatives of even order of the function (5.2.4) are positive and the derivatives of odd order are negative, yield the lower and upper bounds

$$\mathcal{G}_\ell f_\mu \leq \phi(\mu) \leq \mathcal{R}_{\ell+1} f_\mu; \tag{5.3.5}$$

see [10] for details.

We determine a suitable value of μ and an associated approximate solution of (5.1.6) as follows. For $\ell \geq 2$, we seek to solve the nonlinear equation

$$\mathcal{G}_\ell f_\mu = \epsilon^2 \tag{5.3.6}$$

for $\mu > 0$ by Newton's method. One can show similarly as in Section 5.2 that the function $\mu \to \mathcal{G}_\ell f_\mu$ is decreasing and convex. Therefore, assuming that a solution $\mu = \mu_\ell$ of (5.3.6)

exists and that the initial approximate solution $\mu_{\text{init}} \geq 0$ is smaller than μ_ℓ, Newton's method converges quadratically and monotonically to μ_ℓ. If there is no solution, then we increase ℓ. Generally, equation (5.3.6) has a solution already for small values of ℓ.

If the solution μ_ℓ of (5.3.6) satisfies

$$\mathcal{R}_{\ell+1} f_{\mu_\ell} \leq \eta^2 \epsilon^2, \tag{5.3.7}$$

then it follows from (5.3.5) that there is a solution X_{μ_ℓ} of (5.1.6) such that

$$\epsilon \leq \|B - AX_{\mu_\ell}\|_F \leq \eta\epsilon.$$

If (5.3.7) does not hold for μ_ℓ, then we carry out one more GGKB steps and solve (5.3.6) with ℓ replaced by $\ell + 1$. Generally, the bound (5.3.7) can be satisfied already for small values of ℓ, because for any $\mu > 0$, we have generically the inequalities

$$\mathcal{G}_{\ell-1} f_\mu < \mathcal{G}_\ell f_\mu < \phi(\mu) < \mathcal{R}_{\ell+1} f_\mu < \mathcal{R}_\ell f_\mu;$$

see [79] for a proof.

Assume now that (5.3.7) holds for $\mu = \mu_\ell$. We then compute the approximate solution

$$X_{\mu_\ell,\ell} = V_\ell^{(k)}(y_{\mu_\ell} \otimes I_k) \tag{5.3.8}$$

of (5.1.6), where y_{μ_ℓ} solves

$$(\bar{C}_\ell^T \bar{C}_\ell + \mu_\ell^{-1} I_\ell) y = d_1 \bar{C}_\ell^T e_1, \qquad d_1 = \|B\|_F. \tag{5.3.9}$$

These are the normal equations associated with the least-squares problem

$$\min_{y \in \mathbb{R}^\ell} \left\| \begin{bmatrix} \mu_\ell^{1/2} \bar{C}_\ell \\ I_\ell \end{bmatrix} y - d_1 \mu_\ell^{1/2} e_1 \right\|_F. \tag{5.3.10}$$

We compute y_{μ_ℓ} by solving this least-squares problem instead of the normal equations (5.3.9), because this is beneficial numerically, in particular when $\mu_\ell > 0$ is small.

Proposition 5.3.1. *Let μ_ℓ solve* (5.3.6) *and let y_{μ_ℓ} solve* (5.3.10)*. Then the associated approximate solution* (5.3.8) *of* (5.1.6) *satisfies*

$$\|AX_{\mu_\ell,\ell} - B\|_F^2 = \mathcal{R}_{\ell+1} f_{\mu_\ell}.$$

Démonstration. The representation (5.3.8) and (5.3.1) show that

$$AX_{\mu_\ell,\ell} = U_{\ell+1}^{(k)}(\bar{C}_\ell \otimes I_k)(y_{\mu_\ell} \otimes I_k) = U_{\ell+1}^{(k)}(\bar{C}_\ell y_{\mu_\ell} \otimes I_k).$$

Using the above expression gives

$$\begin{aligned}
\|AX_{\mu_\ell,\ell} - B\|_F^2 &= \|U_{\ell+1}^{(k)}(d_1 e_1 \otimes I_k) - U_{\ell+1}^{(k)}(\bar{C}_\ell y_{\mu_\ell} \otimes I_k)\|_F^2 \\
&= \|(d_1 e_1 \otimes I_k) - (\bar{C}_\ell y_{\mu_\ell} \otimes I_k)\|_F^2 \\
&= \|d_1 e_1 - \bar{C}_\ell y_{\mu_\ell}\|_F^2,
\end{aligned}$$

where we recall that $d_1 = \|B\|_F$. We now express y_{μ_ℓ} with the aid of (5.3.9), and apply the identity (5.2.2) with A replaced by $\bar{C}_\ell$, to obtain

$$\begin{aligned}
\|AX_{\mu_\ell,\ell} - B\|_F^2 &= d_1^2 \|e_1 - \bar{C}_\ell(\bar{C}_\ell^T \bar{C}_\ell + \mu_\ell^{-1} I_\ell)^{-1} \bar{C}_\ell^T e_1\|_F^2 \\
&= d_1^2 e_1^T (\mu_\ell \bar{C}_\ell \bar{C}_\ell^T + I_{\ell+1})^{-2} e_1 \\
&= \mathcal{R}_{\ell+1} f_{\mu_\ell}.
\end{aligned}$$

□

The following algorithm outlines the main steps for computing μ_ℓ and $X_{\mu_\ell,\ell}$ that satisfy (5.1.9).

Algorithm 8 The GGKB-Tikhonov method

Input : A, B, k, ϵ, $\eta \geq 1$.

1. Let $U_1 := B/||B||_F$.

2. For $\ell = 1, 2, \ldots$ until $\|AX_{\mu_\ell} - B\|_F \leq \eta\epsilon$

(a) Determine $U_{\ell+1}^{(k)}$ and $V_\ell^{(k)}$, and the bidiagonal matrices C_ℓ and $\bar{C}_\ell$ with GGKB algorithm.

(b) Determine μ_ℓ that satisfies (5.3.6) with Newton's method.

3. Determine y_{μ_ℓ} by solving (5.3.10) and then compute $X_{\mu_\ell,\ell}$ by (5.3.8).

5.4 Golub–Kahan bidiagonalization for problems with multiple right-hand sides

We may consider (5.1.5) as k linear discrete ill-posed problems that have the same matrix A and different right-hand side vectors $b^{(1)}, \ldots, b^{(k)}$; cf. (5.1.10). The solution of linear systems of equations with multiple right-hand sides that may not be known simultaneously and a matrix that stems from the discretization of a well-posed problem has received considerable attention in the literature; see, e.g., [27, 30, 67, 81, 93] and references therein. However, the solution of linear discrete ill-posed problems with multiple right-hand sides that may not be available simultaneously has not. The method described in this section is based on the analysis and numerical experience reported in [48], where it is shown that it often suffices to apply only a few steps of (standard) Golub–Kahan bidiagonalization (GKB) to a matrix A of a linear discrete ill-posed problem to gain valuable information of subspaces spanned by the right and left singular vectors of A

associated with the dominant singular values.

Consider the first system of (5.1.10),

$$Ax^{(1)} = b^{(1)}, \tag{5.4.1}$$

where the right-hand side is the sum of an unknown error-free vector $\widehat{b}^{(1)}$ and an error-vector $e^{(1)}$. Thus, $b^{(1)} = \widehat{b}^{(1)} + e^{(1)}$. A bound $\|e^{(1)}\| \leq \epsilon^{(1)}$ is assumed to be known. Let $\widehat{x}^{(1)}$ denote the first column of the matrix $\widehat{X}$ in (6.1.9). We seek to compute an approximation of $\widehat{x}^{(1)}$ by using (standard) partial Golub–Kahan bidiagonalization (GKB) of A with initial vector $b^{(1)}$.

To explain some properties of the bidiagonalization computed, we introduce the SVD of A,

$$A = W\Sigma Z^T, \tag{5.4.2}$$

where $W, Z \in \mathbb{R}^{n^2 \times n^2}$ are orthogonal matrices and

$$\Sigma = \operatorname{diag}[\sigma_1, \sigma_2, \ldots, \sigma_{n^2}] \in \mathbb{R}^{n^2 \times n^2}, \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_{n^2} = 0.$$

Here r is the rank of A. Let $1 \leq s \leq r$ and let Z_s and W_s consist of the first s columns of Z and W, respectively. Moreover, Σ_s denotes the leading $s \times s$ principal submatrix of Σ. This gives the best rank-s approximation

$$A_s = W_s \Sigma_s Z_s^T \tag{5.4.3}$$

of A in the spectral and Frobenius norms.

The computation of the full SVD (5.4.2) is too expensive for large-scale problems without a particular structure to be practical. The computation of a partial GKB is much

cheaper. Application of ℓ steps of GKB yields the decompositions

$$AV_\ell = U_{\ell+1}\bar{C}_\ell, \quad A^T U_\ell = V_\ell C_\ell^T, \tag{5.4.4}$$

where the matrices $V_\ell = [v_1, v_2, \ldots, v_\ell] \in \mathbb{R}^{n^2\times\ell}$ and $U_{\ell+1} = [u_1, u_2, \ldots, u_{\ell+1}] \in \mathbb{R}^{n^2\times(\ell+1)}$ have orthonormal columns, and U_ℓ consists of the first ℓ columns of $U_{\ell+1}$. Further, $\bar{C}_\ell \in \mathbb{R}^{(\ell+1)\times\ell}$ is lower bidiagonal and C_ℓ is the leading $\ell \times \ell$ submatrix of $\bar{C}_\ell$. We apply reorthogonalization of the columns of $U_{\ell+1}$ and V_ℓ to secure their numerical orthogonality. It is shown in [48] that for sufficiently many steps ℓ, the spaces range$(U_{\ell+1})$ and range(V_ℓ) contain to high accuracy the subspaces range(W_s) and range(Z_s), respectively, for $s \geq 1$ fixed and not too large. Computed examples in [48] indicate that it often suffices to choose $\ell \leq 3s$. This result suggests that we can use the same decomposition (5.4.4) for several right-hand side vectors $b^{(j)}$.

Consider the Tikhonov regularization problem

$$\min_{x\in\text{range}(V_\ell)} \{\|Ax - b^{(1)}\|_2^2 + \mu\|x\|_2^2\} = \min_{y\in\mathbb{R}^\ell} \{\|\bar{C}_\ell y - U_{\ell+1}^T b^{(1)}\|_2^2 + \mu\|y\|_2^2\}, \tag{5.4.5}$$

where $x = V_\ell y$. We determine the regularization parameter $\mu > 0$ so that the computed solution y_μ satisfies the discrepancy principle

$$\|\|\bar{C}_\ell y_\mu - U_{\ell+1}^T b^{(1)}\|_2 = \eta\epsilon^{(1)}. \tag{5.4.6}$$

If no such μ-value exists, then we increase ℓ by one and try to solve (5.4.6) with ℓ replaced by $\ell + 1$ in (5.4.5) and (5.4.6). The small least-squares problem in the right-hand side of (5.4.5) is solved by first expressing it in a form analogous to (5.3.10); see [25] for discussions on the solution of (5.4.5) and on properties of the computed solution. We remark that the vector $U_{\ell+1}^T b^{(1)}$ can be simplified to $e_1\|b^{(1)}\|_2$. The solution y_μ of (5.4.5) determines the approximate solution $x_\mu^{(1)} = V_\ell y_\mu$ of (5.4.1).

We turn to the problem

$$Ax^{(2)} = b^{(2)}, \tag{5.4.7}$$

and compute an approximate solution by solving (5.4.5) with the vector $b^{(1)}$ replaced by $b^{(2)}$. The vector $U_{\ell+1}^T b^{(2)}$ has to be explicitly computed. Therefore it is important that the columns of the matrix $U_{\ell+1}$ be numerically orthonormal. If no $\mu > 0$ can be determined so that (5.4.6) can be satisfied with $b^{(1)}$ replaced by $b^{(2)}$, then we carry out one more step of Golub–Kahan bidiagonalization (5.4.4) ; otherwise we compute the solution y_μ of (5.4.5) with the available decomposition.

Let μ be such that the discrepancy principle holds. Then we obtain the approximate solution $x_\mu^{(2)} = V_\ell y_\mu$ of (5.4.7). We proceed in the same manner to solve $Ax^{(i)} = b^{(i)}$ for $i = 3, 4, \ldots, k$.

Algorithm 9 The GKB-Tikhonov method

Input : A, k, $b^{(1)}, b^{(2)}, \ldots, b^{(k)}$, $\epsilon^{(1)}, \epsilon^{(2)}, \ldots, \epsilon^{(k)}$, $\eta \geq 1$.

1. Let $u_1 := b^{(1)}/\|b^{(1)}\|_2$.

2. Compute $AV_\ell = U_{\ell+1}\bar{C}_\ell, \quad A^T U_\ell = V_\ell C_\ell^T$

3. For $i = 1, 2, \ldots, k$

(a) Compute $\min\limits_{y_\mu \in \mathbb{R}^\ell} \{\|\bar{C}_\ell y_\mu - U_{\ell+1}^T b^{(i)}\|_2^2 + \mu \|y_\mu\|_2^2\}$

(b) If $\|\bar{C}_\ell y_\mu - U_{\ell+1}^T b^{(i)}\|_2 > \eta \epsilon^{(i)}$

i. $\ell := \ell + 1$

ii. Return to step 5.

(c) Compute $x_\mu^{(i)} = V_\ell y_\mu$

We will compare this algorithm and Algorithms 7 and 8 to the following "trivial"

method that is based on solving each one of the linear discrete ill-posed problems (5.1.10) independently with the aid of (standard) Golub–Kahan bidiagonalization. Thus, we apply Algorithm 7 with block size one to each one of the k linear discrete ill-posed problems (5.1.10) independently. We refer to this scheme as GKB. We expect it to require the most matrix-vector product evaluations of the methods in our comparison, because we compute a new partial standard Golub–Kahan bidiagonalization for each one of the vectors $b^{(j)}$, $j = 1, \ldots, k$. Moreover, this method does not benefit from the fact that on many modern computers the evaluation of matrix-block-vector products with a large matrix A does not require much more time than the evaluation of a matrix-vector product with a single vector for small block sizes; see, e.g., [40] for discussions on this and related issues.

5.5 Numerical results

This section provides some numerical results to show the performance of Algorithms 7-9 and GKB when applied to the solution of linear discrete ill-posed problems with the same matrix and different right-hand sides. The first example applies these algorithms to the solution of linear discrete ill-posed problems with several right-hand sides defined by matrices that stem from Regularization Tools by Hansen [61], while the second example discusses the restoration of RGB images that have been contaminated by within-channel blur and noise. All computations were carried out using the MATLAB environment on an Pentium(R) Dual-Core CPU T4200 computer with 4 GB of RAM. The computations were done with approximately 15 decimal digits of relative accuracy.

Example 1

We would like to solve linear discrete ill-posed problems (5.1.10) with the matrix $A \in \mathbb{R}^{70^2 \times 70^2}$ determined by the functions phillips in Regularization Tools [61]. The matrix is a discretization of a Fredholm integral equation of the first kind that describes a convolution. The function phillips also determines the error-free data vector $\widehat{b}^{(1)} \in \mathbb{R}^{70^2}$ and the associated error-free solution $\widehat{x}^{(1)} \in \mathbb{R}^{70^2}$. The other error-free data vectors $\widehat{b}^{(i)} \in \mathbb{R}^{70^2}$, $i = 2, \ldots, k$, are obtained by setting $\widehat{x}^{(i)} = \widehat{x}^{(i-1)} + y/2$ for $i = 2, \ldots, k$, where y is a vector obtained by discretization of function of the form $\alpha \cos(\beta t) + \gamma$, where α, β, and γ are scalars. For this example we let $\alpha = 1/2$, $\beta = 1/3$ and $\gamma = 1/4$. The error-free right hand sides are obtained by letting $\widehat{b}^{(i)} = A\widehat{x}^{(i)}$ for $i = 2, \ldots, k$, in (5.1.10). The error-vectors $e^{(i)}$ are scaled to correspond to a specified noise level. This is simulated with

$$e^{(i)} := \widetilde{\delta} \|\widehat{b}^{(i)}\|_2 \widetilde{e}^{(i)},$$

where $\widetilde{\delta}$ is the noise level, and the vector $\widetilde{e}^{(i)} \in \mathbb{R}^{70^2}$ has normally distributed random entries with mean zero and variance one.

When the data vectors $b^{(i)}$, $i = 1, \ldots, k$, are available sequentially, the linear discrete ill-posed problems (5.1.10) can be solved one by one by Algorithms 9 or GKB. If the data vectors are available simultaneously, then Algorithms 7 and 8 also can be used to solve (5.1.10). The latter algorithms require that the noise level for each discrete ill-posed problem (5.1.10) is about the same. This is a reasonable assumption for many applications.

Table 5.1 compares the number of matrix-vector product evaluations and the CPU time required by Algorithm 7-9 and GKB for $k = 10$ and noise-contaminated data vectors $b^{(i)}$ corresponding to the noise levels $\widetilde{\delta} = 10^{-2}$ and $\widetilde{\delta} = 10^{-3}$. For the discrepancy principle,

we chose $\eta = 1.1$. The displayed relative error in the computed solutions is the maximum error for each one of the k problems (5.1.10). The number of matrix-vector products (MVP) shown is the number of matrix-vector product evaluations with A and A^T with a single vector. Thus, each iteration step of Algorithms 7 and 8 adds $2k$ matrix-vector product evaluations to the count. The number of matrix-vector product evaluations does not give an accurate idea of the computing time required. We therefore also present timings for the algorithms. The Tables 5.2 and 5.3 are analogous to Table 5.1. They differ from the latter in that the matrix $A \in \mathbb{R}^{70^2 \times 70^2}$ is determined by the function baart for Table 5.2 and the function shaw for Table 5.3. The vectors $b^{(i)}$ are determined analogously as for the phillips test problem. Both functions are from [61] and compute discretizations of Fredholm integral equations of the first kind.

Noise level	Method	MVP	Relative error	CPU-time (sec)
10^{-3}	Algorithm 7	100	1.46×10^{-2}	3.87
	Algorithm 8	200	1.31×10^{-2}	7.63
	Algorithm 9	16	2.28×10^{-2}	1.52
	GKB	162	1.43×10^{-2}	13.22
10^{-2}	Algorithm 7	80	2.54×10^{-2}	3.08
	Algorithm 8	120	2.61×10^{-2}	4.67
	Algorithm 39	10	2.52×10^{-2}	1.01
	GKB	140	2.60×10^{-2}	11.50

TABLE 5.1 – Results for the phillips test problem.

Noise level	Method	MVP	Relative error	CPU-time (sec)
10^{-3}	Algorithm 7	40	4.27×10^{-2}	1.61
	Algorithm 8	80	5.62×10^{-2}	3.31
	Algorithm 9	8	5.20×10^{-2}	0.81
	GKB	80	5.46×10^{-2}	7.11
10^{-2}	Algorithm 7	40	5.02×10^{-2}	1.51
	Algorithm 8	60	7.36×10^{-2}	2.57
	Algorithm 9	8	5.77×10^{-2}	0.83
	GKB	62	6.78×10^{-2}	5.63

TABLE 5.2 – Results for the baart test problem.

Noise level	Method	MVP	Relative error	CPU-time (sec)
10^{-3}	Algorithm 7	100	5.20×10^{-2}	3.88
	Algorithm 8	200	4.42×10^{-2}	7.57
	Algorithm 9	14	3.98×10^{-2}	1.30
	GKB	184	4.72×10^{-2}	14.93
10^{-2}	Algorithm 7	40	1.82×10^{-1}	1.52
	Algorithm 8	100	1.55×10^{-1}	4.36
	Algorithm 9	10	1.27×10^{-1}	0.98
	GKB	100	1.55×10^{-1}	8.84

TABLE 5.3 – Results for the shaw test problem.

Tables 5.1-5.3 show Algorithm 9 to require the fewest matrix-vector product evaluations and to give approximate solutions of comparable or higher quality than the other algorithms. Algorithms 7 and 8 require about the same number of matrix-vector product evaluations, but the former algorithm demands less CPU time because it implements a block method.

Example 2

This example illustrates the performance of Algorithms 7-9 and GKB when applied to the restoration of 3-channel RGB color images that have been contaminated by blur and noise. The corrupted image is stored in a block vector B with three columns. The desired (and assumed unavailable) image is stored in the block vector $\widehat{X}$ with three columns. The blur-contaminated, but noise-free image associated with $\widehat{X}$, is stored in the block vector $\widehat{B}$. The block vector E represents the noise in B, i.e., $B := \widehat{B} + E$. We define the noise level

$$\nu = \frac{||E||_F}{||\widehat{B}||_F}.$$

To determine the effectiveness of our solution methods, we evaluate the relative error

$$\text{Relative error} = \frac{||\widehat{X} - X_{\mu_\ell}||_F}{||\widehat{X}||_F},$$

where X_{μ_ℓ} denotes the computed restoration.

We consider the within-channel blurring only. Hence the blurring matrix A_3 in (5.1.4) is the 3×3 identity matrix. The blurring matrix A in (5.1.4), which describes the blurring within each channel, models Gaussian blur and is determined by the following Gaussian

PSF,

$$h_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\big(-\frac{x^2+y^2}{2\sigma^2}\big).$$

The blurring matrix A is a symmetric block Toeplitz matrix with Toeplitz blocks. It is generated with the MATLAB function `blur` from [61]. This function has two parameters, the half-bandwidth of the Toeplitz blocks r and the variance σ of the Gaussian PSF. For this example we let $\sigma = 2$ and $r = 4$. The original (unknown) RGB image $\widehat{X} \in 256\times256\times3$ is the `papav256` image from MATLAB. It is shown in the left-hand side of Figure 5.1. The associated blurred and noisy image $B = A\widehat{X} + E$ is shown in the right-hand side of the figure. The noise level is $\nu = 10^{-3}$. Given the contaminated image B, we would like to recover an approximation of the original image $\widehat{X}$. Table 5.4 compares the number of matrix-vector product evaluations, the computing time, and the relative errors in the computed restorations. We use the discrepancy principle with $\eta = 1.1$ to determine the regularization parameter.

The restoration obtained with Algorithm 7 for noise level $\nu = 10^{-3}$ is shown in the left-hand side of Figure 5.2. The discrepancy principle for this algorithm and this noise level is satisfied after $\ell = 70$ steps of the BGKB method. This corresponds to $3 \times 2 \times 70$ matrix-vector product evaluations.

The restoration determined by Algorithm 7 is shown in the right-hand side of Figure 5.2. The GGKB method requires $\ell = 66$ steps to satisfy the discrepancy principle. Algorithm 8 is the fastest, but yields restorations of lower quality than the other algorithms for this example.

TABLE 5.4 – Results for Example 2.

Noise level	Method	MVP	Relative error	CPU-time (sec)
10^{-3}	Algorithm 7	420	4.44×10^{-2}	9.52
	Algorithm 8	396	4.46×10^{-2}	6.30
	Algorithm 9	68	2.81×10^{-1}	3.76
	GKB	628	4.16×10^{-2}	15.02
10^{-2}	Algorithm 7	102	6.72×10^{-2}	1.88
	Algorithm 8	90	6.71×10^{-2}	1.38
	Algorithm 9	18	2.88×10^{-1}	0.77
	GKB	124	6.57×10^{-2}	2.53

FIGURE 5.1 – Example 2 : Original image (left), blurred and noisy image (right).

FIGURE 5.2 – Example 2 : Restored image by Algorithm 7 (left), restored image by Algorithm 8 (right).

Chapitre 6

A genaralized Krylov subspace method for TV regularization

6.1 Introduction

In this chapter we consider the solution of the following matrix equation

$$B = H_2 X H_1^T, \tag{6.1.1}$$

where B is generally contaminated by noise. H_1 and H_2 are matrices of ill-determined rank, which makes the solution X very sensitive to perturbations in B. Discrete ill-posed problems of the form (6.1.1) arise, for instance, from the discretization of Fredholm integral equations of the first kind in two space-dimensions,

$$\int\int_{\Omega} K(x, y, s, t) f(s, t) ds dt = g(x, y), \qquad (x, y) \in \Omega', \tag{6.1.2}$$

where Ω and Ω' are rectangles in $\mathbb{R}^2$ and the kernel is separable

$$K(x, y, s, t) = k_1(x, s) k_2(y, t), \quad (x, y) \in \Omega', \quad (s, t) \in \Omega,$$

The aim of this chapter is to solve this problem with application to one single channel and multichannel images. For single channel images we seek to recover an unknown vector from limited information. This problem is mathematically formulated as the following model

$$b = Hx \tag{6.1.3}$$

where $x \in \mathbb{R}^{mn}$ is a vector denoting the unknown solution, $b \in \mathbb{R}^{mn}$ is a vector denoting the observed data contaminated by noise and $H \in \mathbb{R}^{mn \times mn}$ is a linear map. The problem arises, for instance in image restoration [2, 6, 18, 62, 64]. In this chapter we focus on the solution of (6.1.1) with application to image restoration in which x represents the unknown sharp image that is to be estimated from its blurry and noisy observation b. The matrix H is the blurring operator characterized by a PSF describing this blur. Due to the ill-conditioning of the matrix H and the presence of the error e, the problem (6.1.1) cannot be easily solved which mean that minimization of only the fidelity term typically yields a meaningless computed solution. Therefore, to stabilize the recovered image, regularization is needed. There are several techniques to regularize the linear inverse problem given by equation (6.1.1) ; see, for example, [96, 43, 88, 98]. All of these techniques stabilize the restoration process by adding a regularization term, depending on some priori knowledge about the unknown image, resulting in the model

$$\min_x \{\|Hx - b\|_p^p + \mu \|\Phi(x)\|_q^q\}. \tag{6.1.4}$$

where $\Phi(x)$ is the regularizer that enforces the priori knowledge and the parameter μ is used to balance the two terms. This problem is referred to as $\ell_p - \ell_q$ minimization problem. Different choices of $\Phi(x)$, p and q lead to a wide variety of regularizers. Among them we find the well known Tikhonov regularization, where Φ is the identity matrix ($\Phi = I$),

$p = 2$ and $q = 2$, see for example [96]. If the goal is to enforce sparsity on the solution, one can also consider $\Phi = I$, $p = 2$ and $q = 1$. Another well-known class of regularizers are based on total variation (TV), which is a better choice if the goal is to preserve sharp edges. In this case one let Φ to be the discrete gradient operator, see [88]. The problem (6.1.4) has been studied in many papers to propose nonlinear optimization algorithms that can deal with the nonlinear properties of this problem ; see for example [97, 100]. These techniques are computationally demanding if the main cost of computation is the matrix-vector multiplication (MVM). It is our main goal to recover a good approximation of the unknown sharp image at low computational cost. Because of some unique features in images, we seek an image restoration algorithm that utilizes blur information, exploits the spatially invariant properties. For this reason we suppose that the PSF is identical in all parts of the image and separates into horizontal and vertical components. Then the matrix H is a tensor of two matrices H_1 and H_2. By using the properties (1.2.3), the equation (6.1.3) can be rewritten as

$$B = H_2 X H_1^T, \tag{6.1.5}$$

where $X = \mathsf{mat}(x)$ and $B = \mathsf{mat}(b)$, Which yields the model (6.1.1). Recovering multi-channel images from their blurry and noisy observations can be seen as a linear system of equations with multiple right-hand sides. The most commonly multichannel images is the RGB representation, which uses three channels ; see [44, 62]. It should be pointed out that the algorithms proposed in this chapter can be applied to the solution of Fredholm integral equations of the first kind in two or more space dimensions and to the restoration of hyper-spectral images. The latter kind of images generalize color images in that they

allow more than three "colors" ; see, e.g., [77]. If the channels are represented by $m \times n$ pixels, the full blurring model is described by the following form

$$b = Hx, \tag{6.1.6}$$

where b and x in $\mathbb{R}^{kmn}$, represent the blurred and noisy multichannel image and the original image respectively. For an image with k channels, they are given by

$$b = [b^{(1)}; b^{(2)}; ...; b^{(k)}], \quad x = [x^{(1)}; x^{(2)}; ...; x^{(k)}],$$

where $b^{(i)}$ and $x^{(i)}$ in $\mathbb{R}^{mn}$ are obtained by stacking the columns of each channel on top of each other. The $kmn \times kmn$ multichannel blurring matrix H is given by

$$H = H_1 \otimes H_2, \tag{6.1.7}$$

The matrix $H_2 \in \mathbb{R}^{mn \times mn}$ represents the same within-channel blurring in all the k channels. The matrix H_1 of dimension $k \times k$ models the cross-channel blurring, which is the same for all pixels in the case of a spatially invariant blur. If $H_1 = I$, the blurring is said to be within-channel. If no colour blurring arises (i.e., $H_1 = I$), then k independent deblurring problems are solved ; hence the spatially invariant blurring model is given by

$$b_i = H_2 x_i, \quad i = 1, ..., k. \tag{6.1.8}$$

In this case, the goal is to model the blurring of k channels image as a linear system of equations with k right-hand sides. For this reason we let B and X in $\mathbb{R}^{mn \times k}$ to be denoted by $\left[b^{(1)}, b^{(2)}, ..., b^{(k)}\right]$ and $\left[x^{(1)}, x^{(2)}, ..., x^{(k)}\right]$, respectively. The optical blurring is then modeled by

$$B = H_2 X, \tag{6.1.9}$$

which yields the model (6.1.1) with $H_1 = I$. When the spatially invariant cross-channel is present (i.e., $H_1 \neq I$) and by using the Kronecker product properties, the following blurring model is to be solved

$$B = H_2 X H_1^T, \tag{6.1.10}$$

which also yields the model (6.1.1). Introduce the linear operator

$$\begin{aligned} \mathcal{H} : \mathbb{R}^{p\times q} &\rightarrow \mathbb{R}^{p\times q} \\ \mathcal{H}(X) &= H_2 X H_1^T. \end{aligned}$$

Its transpose is given by $\mathcal{H}^T(X) = H_2^T X H_1$. The problem (6.1.1) can be then expressed as

$$B = \mathcal{H}(X).$$

The total variation regularization is known to be the most popular and effective techniques for the images restoration. Given an image defined as a function $u : \Omega \longrightarrow \mathbb{R}$, where Ω is a bounded open subset of $\mathbb{R}^2$, the total variation (TV) of u can be defined as

$$\mathrm{TV}_k(u) = \int_\Omega \|\nabla u(x)\|_k dx, \tag{6.1.11}$$

where ∇ denotes the gradient of u and $\|.\|_k$ is a norm in $\mathbb{R}^2$. When u is represented by $m \times n$ image X, a discrete form of (6.1.11) is always used, given by

$$\mathrm{TV}_1(X) = \sum_{i=1}^{m}\sum_{j=1}^{n}\Big(|\,(D_{1,n}X)_{ij}\,| + |\,(D_{1,m}X)_{ij}\,|\Big) \tag{6.1.12}$$

in the anisotropic total variation case, or

$$\mathrm{TV}_2(X) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sqrt{\Big((D_{1,n}X)_{ij}^2 + (D_{1,m}X)_{ij}^2\Big)} \tag{6.1.13}$$

in the isotropic total variation case. $D_{1,m}$ and $D_{1,n}$ denote the finite difference approximations of the horizontal and vertical first derivative operators, respectively, and they are defined as follows

$$\begin{pmatrix} D_{1,n} \\ D_{1,m} \end{pmatrix} X = \begin{pmatrix} CX \\ XC^T \end{pmatrix}, \tag{6.1.14}$$

where

$$C := \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{d-1\times d},$$

where d is the number of pixels in each row and column of the image considered. For the ill-posed image restoration problem (6.1.1), the resulting matrices H_1 and H_2 are ill-conditioned. By regularization of the problem (6.1.1), we solve as a special case one of the following matrix problems :

$$\min_X \left(\|\mathcal{H}(X) - B\|_F^2 + \mu \mathrm{TV}_k(X) \right), \quad k = 1, 2 \tag{6.1.15}$$

or

$$\min_X \left(\|\mathcal{H}(X) - B\|_{1,1} + \mu \mathrm{TV}_k(X) \right) \quad k = 1, 2. \tag{6.1.16}$$

where $\|.\|_{1,1}$ is the ℓ_1 norm. Problems (6.1.15) and (6.1.16) are refereed to as TV/L2 and TV/L1 minimization, respectively.

6.2 TV/L2 minimization problem

In this section we consider the solution of the following TV/L2 minimization problem

$$\min_X \left(\|\mathcal{H}(X) - B\|_F^2 + \mu \mathrm{TV}_2(X) \right). \tag{6.2.1}$$

The model (6.2.1) is very difficult to solve directly due to the non-differentiability and non-linearity of the TV term. It is our goal to develop an efficient TV minimization scheme to handle this problem. The core idea is based on augmented Lagrangian method (ALM) [59, 85] and alternating direction method (ADM) [53]. The idea of ALM is to transform the unconstrained minimization task (6.2.1) into an equivalent constrained optimization problem, and then add a quadratic penalty term instead of the constraint violation with the multipliers. The idea of ADM is to decompose the transformed minimization problem into three easier and smaller subproblems such that some involved variables can be minimized separately and alternatively. Let us begin by considering the equivalent equality-constrained problem of (6.2.1). We first notice that the minimization problem (6.2.1) can be rewritten as

$$\begin{aligned} \min_{X,M^{(n)},M^{(m)}} & \left(\|\mathcal{H}(X) - B\|_F^2 + \mu \sum_{i=1}^{m} \sum_{j=1}^{n} \|M_{i,j}\|_2 \right), \\ \text{subject to} \quad & D_{1,n}X = M^{(n)}, \quad D_{1,m}X = M^{(m)}. \end{aligned} \tag{6.2.2}$$

where $M_{i,j} = \left[(D_{1,n}X)_{ij}, (D_{1,m}X)_{ij} \right]$. If we set $M_{i,j}^{(n)} = (D_{1,n}X)_{ij}$ and $M_{i,j}^{(m)} = (D_{1,m}X)_{ij}$ This constrained problem can be also formulated as

$$\begin{aligned} \min \quad & F(X) + G(Y), \\ \text{subject to} \quad & DX = Y, \end{aligned} \tag{6.2.3}$$

where,

$$F(X) = \|\mathcal{H}(X) - B\|_F^2, \quad G(Y) = \mu \sum_{i=1}^{m} \sum_{j=1}^{n} \|M_{i,j}\|_2, \quad D = \begin{pmatrix} D_{1,n} \\ \\ D_{1,m} \end{pmatrix}, \quad Y = \begin{pmatrix} M^{(n)} \\ \\ M^{(m)} \end{pmatrix}$$

The augmented Lagrangian function of (6.2.3) is defined as

$$\mathcal{L}_\beta (X, Y, Z) = F(X) + G(Y) + \langle DX - Y, Z \rangle + \frac{\beta}{2} \|DX - Y\|_F^2, \tag{6.2.4}$$

where $Z \in \mathbb{R}^{2m\times n}$ is the Lagrange multiplier of the linear constraint and $\beta > 0$ is the penalty parameter for the violation of this linear constraint.

To solve the nonlinear problem (6.2.1), we find the saddle point of the Lagrangian (6.2.4) by using the ADM method. The idea of this method is to apply an alternating minimization iterative procedure, namely, for $k = 0, 1, ...$, we solve

$$(X_{k+1}, Y_{k+1}) = \arg\min_{X} \mathcal{L}_\beta(X, Y, Z_k). \tag{6.2.5}$$

The Lagrange multiplier is updated by

$$Z_{k+1} \quad = \quad Z_k + \beta\left(DX_{k+1} - Y_{k+1}\right). \tag{6.2.6}$$

6.2.1 Solving the Y-problem

Given X, Y_{k+1} can be obtained by solving

$$\min_{Y} \mu\sum_{i=1}^{m}\sum_{j=1}^{n}\|M_{i,j}\|_2 + \frac{\beta}{2}\|DX - Y\|_F^2 + \langle DX - Y, Z_k\rangle_F, \tag{6.2.7}$$

which is equivalent to solve

$$\min_{Y} \mu\sum_{i=1}^{m}\sum_{j=1}^{n}\|M_{i,j}\|_2 + \frac{\beta}{2}\left\|\begin{pmatrix} M^{(n)} \\ \\ M^{(m)}\end{pmatrix} - \begin{pmatrix} D_{1,n}X \\ \\ D_{1,m}X\end{pmatrix} - \frac{1}{\beta}\begin{pmatrix} Z_k^{(1)} \\ \\ Z_k^{(2)}\end{pmatrix}\right\|_F^2, \tag{6.2.8}$$

which is also equivalent to solve the so-called M-subproblem

$$\min_{M_{i,j}} \sum_{i=1}^{m}\sum_{j=1}^{n}\mu\|M_{i,j}\|_2 + \frac{\beta}{2}\left|M_{ij}^{(n)} - K_{ij}\right|_F^2 + \frac{\beta}{2}\left|M_{ij}^{(m)} - L_{ij}\right|_F^2 \tag{6.2.9}$$

where $K_{ij} = (D_{1,n}X)_{ij} + \frac{1}{\beta}\left(Z_k^{(1)}\right)_{ij}$ and $L_{ij} = (D_{1,m}X)_{ij} + \frac{1}{\beta}\left(Z_k^{(2)}\right)_{ij}$. To solve (6.2.9) we use following well-known two dimensional shrinkage formula [75]

$$\textbf{Shrink}(y, \gamma, \delta) = \max\left\{\left\|y + \frac{\gamma}{\delta}\right\|_2 - \frac{1}{\delta}, 0\right\}\frac{y + \gamma/\delta}{\|y + \gamma/\delta\|_2}, \tag{6.2.10}$$

where the convention $0\cdot(0/0) = 0$ is followed. The solution of (6.2.9) is then given by

$$M_{i,j} = \max\left\{\|T_{i,j}\|_2 - \frac{\mu}{\beta}, 0\right\}\frac{T_{i,j}}{\|T_{i,j}\|_2}, \tag{6.2.11}$$

where $T_{i,j} = \left[(D_{1,n}X_k)_{i,j} + \frac{1}{\beta}\left(Z_k^{(1)}\right)_{i,j}, (D_{1,m}X_k)_{i,j} + \frac{1}{\beta}\left(Z_k^{(2)}\right)_{i,j}\right]$.

For the anisotropic case we solve the following problem

$$\min_{M_{i,j}} \sum_{i=1}^{m}\sum_{j=1}^{n} \mu\|M_{i,j}\|_1 + \frac{\beta}{2}\left|M_{ij}^{(n)} - K_{ij}\right|_F^2 + \frac{\beta}{2}\left|M_{ij}^{(m)} - L_{ij}\right|_F^2, \tag{6.2.12}$$

which can be also solved by the one dimensional shrinkage formula. This gives

$$M_{ij}^{(n)} = \max\left\{K_{ij} - \frac{\mu}{\beta}, 0\right\}.\text{sign}\left(K_{ij}\right), \tag{6.2.13}$$

$$M_{ij}^{(m)} = \max\left\{L_{ij} - \frac{\mu}{\beta}, 0\right\}.\text{sign}\left(L_{ij}\right). \tag{6.2.14}$$

6.2.2 Solving the X-problem

Given Y, X_{k+1} can be obtained by solving

$$\min_X \frac{\beta}{2}\|DX - Y\|_F^2 + \langle DX - Y, Z_k\rangle_F + \|\mathcal{H}(X) - B\|_F^2. \tag{6.2.15}$$

This problem can be also solved by considering the following normal equation

$$H_1^T H_1 X H_2^T H_2 + \beta D^T DX = H_1^T B H_2 + D^T\left(\beta Y - Z_k\right). \tag{6.2.16}$$

The linear matrix equation can be rewritten in the following form

$$A_1 X A_2 + A_3 X A_4 = E_k, \quad k = 1, ..., \tag{6.2.17}$$

where $A_1 = H_1^T H_1$, $A_2 = H_2^T H_2$, $A_3 = \beta D^T D$, $A_4 = I$ and $E_k = H_1^T B H_2 + D^T\left(\beta Y - Z_k\right)$. The equation (6.2.17) is refereed to as the generalized Sylvester matrix equation. We will see in section 6.4 how to compute approximate solutions to those matrix equations

6.2.3 Convergence analysis of TV/L2 problem

For the vector case, many convergence results have been proposed in the literature ; see for instance [41, 65]. For completeness, we give a proof here for the matrix case. A function Ψ is said to be proper if the domain of Ψ denoted by $\mathbf{dom}\Psi := \{U \in \mathbb{R}^{p\times q}, \Psi(U) < \infty\}$ is not empty. For the problem (6.2.3), F and G are closed proper convex functions. According to [35, 89], the problem (6.2.3) is solvable, i.e., there exist X_* and Y_*, not necessarily unique that minimize (6.2.3). Let $\mathcal{W} = \Omega \times \mathcal{Y} \times \mathbb{R}^{p\times q}$, where Ω and $\mathcal{Y}$ are given closed and convex nonempty sets. The saddle-point problem is equivalent to finding $(X_*, Y_*, Z_*) \in \mathcal{W}$ such that

$$\mathcal{L}_\beta(X_*, Y_*, Z) \leq \mathcal{L}_\beta(X_*, Y_*, Z_*) \leq \mathcal{L}_\beta(X, Y, Z_*), \qquad \forall\, (X, Y, Z) \in \mathcal{W}. \tag{6.2.18}$$

The properties of the relation between the saddle-points of $\mathcal{L}_\beta$ and $\mathcal{L}_0$ and the solution of (6.2.3) are stated by the following theorem from [41]

Theorem 6.2.1. *(X_*, Y_*, Z_*) is a saddle-point of $\mathcal{L}_0$ if and only if (X_*, Y_*, Z_*) is a saddle-point of $\mathcal{L}_\beta$ $\forall \beta > 0$. Moreover (X_*, Y_*) is a solution of (6.2.3).*

We will see in what follows how this theorem can be used to give the convergence of (X_{k+1}, Y_{k+1}). It should be pointed out that the idea of our proof follows the convergence results in [16].

Theorem 6.2.2. *Assume that (X_*, Y_*, Z_*) is a saddle-point of $\mathcal{L}_\beta$ $\forall \beta > 0$. The sequence $(X_{k+1}, Y_{k+1}, Z_{k+1})$ generated by Algorithm 10 satisfies*

1. $\lim\limits_{k\to+\infty} F(X_{k+1}) + G(Y_{k+1}) = F(X_*) + G(Y_*)$,

2. $\lim_{k\to+\infty} \|DX_{k+1} - Y_{k+1}\|_F = 0,$

Proof In order to show the convergence of this theorem, it suffice to show that the non-negative function

$$F^k = \frac{1}{\beta}\|Z_k - Z_*\|_F^2 + \beta\|X_k - X_*\|_F^2 \tag{6.2.19}$$

decreases at each iteration. Let us define S_k, M_k and M_* as

$$S_k = DX_k - Y_k, \quad M_k = F(X_k) + G(Y_k), \quad M_* = F(X_*) + G(Y_*).$$

In the following we show

$$F^{k+1} \leq F^k - \beta\|S_{k+1}\|_F^2 - \beta\|Y_{k+1} - Y_k\|_F^2. \tag{6.2.20}$$

Since (X_*, Y_*, Z_*) is a saddle-point of $\mathcal{L}_\beta \ \forall\beta > 0$, it follows from Theorem 6.2.1 that (X_*, Y_*, Z_*) is also a saddle-point of $\mathcal{L}_0$. This is characterized by

$$\mathcal{L}_0(X_*, Y_*, Z) \leq \mathcal{L}_0(X_*, Y_*, Z_*) \leq \mathcal{L}_0(X, Y, Z_*), \qquad \forall\,(X, Y, Z) \in \mathcal{W}. \tag{6.2.21}$$

From the second inequality of (6.2.21), we have

$$M_* - M_{k+1} \leq \langle S_{k+1}, Z_*\rangle_F\,. \tag{6.2.22}$$

Since X_{k+1} is a minimizer of $\mathcal{L}_\beta \ \forall\beta > 0$, the optimality conditions reads

$$2\mathcal{H}^T\left(\mathcal{H}(X_{k+1}) - B\right) + D^T(Z_k + \beta(DX_{k+1} - Y_k)) = 0. \tag{6.2.23}$$

By plugging $Z_k = Z_{k+1} - \beta(DX_{k+1} - Y_{k+1})$ and rearranging we obtain

$$2\mathcal{H}^T\left(\mathcal{H}(X_{k+1}) - B\right) + D^T(Z_{k+1} - \beta(Y_{k+1} - Y_k)) = 0, \tag{6.2.24}$$

which means that X_{k+1} minimizes

$$F(X) + \langle Z_{k+1} + \beta(Y_{k+1} - Y_k), DX \rangle_F . \tag{6.2.25}$$

It follows that

$$F(X_{k+1}) - F(X_*) \leq \langle Z_{k+1} + \beta(Y_{k+1} - Y_k), DX_* \rangle_F - \langle Z_{k+1} + \beta(Y_{k+1} - Y_k), DX_{k+1} \rangle_F . \tag{6.2.26}$$

A similar argument shows that

$$G(Y_{k+1}) - G(Y_*) \leq \langle Z_{k+1}, Y_{k+1} \rangle - \langle Z_{k+1}, Y_* \rangle_F . \tag{6.2.27}$$

Adding (6.2.26) and (6.2.27) and using $DX_* = Y_*$ implies

$$M_{k+1} - M_* \leq - \langle S_{k+1}, Y_{k+1} \rangle_F - \langle \beta(Y_{k+1} - Y_k), S_{k+1} + (Y_{k+1} - Y_*) \rangle_F . \tag{6.2.28}$$

Adding (6.2.22) and (6.2.28) gives

$$2 \langle S_{k+1}, Z_{k+1} - Z_k \rangle_F + 2 \langle \beta(Y_{k+1} - Y_k), S_{k+1} \rangle_F + 2 \langle \beta(Y_{k+1} - Y_k), (Y_{k+1} - Y_*) \rangle_F \leq 0 \tag{6.2.29}$$

Substituting $Z_{k+1} = Z_k + \beta S_{k+1}$ gives

$$2 \langle S_{k+1}, Z_{k+1} - Z_k \rangle_F = 2 \langle S_{k+1}, Z_k - Z_* \rangle_F + \beta \|S_{k+1}\|_F^2 + \beta \|S_{k+1}\|_F^2. \tag{6.2.30}$$

Since $S_{k+1} = \frac{1}{\beta} (Z_{k+1} - Z_k)$, it follows

$$2 \langle S_{k+1}, Z_k - Z_* \rangle_F + \beta \|S_{k+1}\|_F^2 = \frac{2}{\beta} \langle Z_{k+1} - Z_k, Z_k - Z_* \rangle_F + \frac{1}{\beta} \|Z_{k+1} - Z_k\|_F^2 \tag{6.2.31}$$

Substituting $Z_{k+1} - Z_k = (Z_{k+1} - Z_*) - (Z_k - Z_*)$, the right hand side of (6.2.31) can be written as

$$\frac{1}{\beta} \left(\|Z_{k+1} - Z_*\|_F^2 - \|Z_k - Z_*\|_F^2 \right) . \tag{6.2.32}$$

Substituting $Y_{k+1} - Y_* = (Y_{k+1} - Y_k) + (Y_{k+1} - Y_*)$ shows that

$\beta\|S_{k+1}\|_F^2 + 2\langle\beta(Y_{k+1} - Y_k), S_{k+1}\rangle_F + 2\langle\beta(Y_{k+1} - Y_k), (Y_{k+1} - Y_*)\rangle_F$ can be expressed as

$$\beta\|S_{k+1} + (Y_{k+1} - Y_k)\|_F^2 + \beta\|Y_{k+1} - Y_k\|_F^2 + 2\beta\langle Y_{k+1} - Y_k, Y_k - Y_*\rangle_F . \tag{6.2.33}$$

Substituting $Y_{k+1} - Y_k = (Y_{k+1} - Y_*) - (Y_k - Y_*)$ in the last two terms shows that (6.2.33) can be expressed as

$$\beta\|S_{k+1} + (Y_{k+1} - Y_k)\|_F^2 + \beta\left(\|Y_{k+1} - Y_*\|_F^2 - \|Y_k - Y_*\|_F^2\right) \tag{6.2.34}$$

Using (6.2.32) and (6.2.34) shows that (6.2.29) can be expressed as

$$F^k - F^{k+1} \geq \beta\|S_{k+1} + (Y_{k+1} - Y_k)\|_F^2. \tag{6.2.35}$$

To show (6.2.20), it is now suffice to show that $2\beta\langle S_{k+1}, Y_{k+1} - Y_k\rangle_F \geq 0$. Since (X_k, Y_k, Z_k) and $(X_{k+1}, Y_{k+1}, Z_{k+1})$ are also minimizers of $\mathcal{L}_\beta$, we have as in (6.2.27)

$$G(Y_{k+1}) - G(Y_k) \leq \langle Z_{k+1}, Y_{k+1}\rangle_F - \langle Z_{k+1}, Y_k\rangle_F , \tag{6.2.36}$$

and

$$G(Y_k) - G(Y_{k+1}) \leq \langle Z_k, Y_k\rangle_F - \langle Z_k, Y_{k+1}\rangle_F . \tag{6.2.37}$$

By addition,

$$\langle Y_{k+1} - Y_k, Z_{k+1} - Z_k\rangle \geq 0. \tag{6.2.38}$$

Substituting $Z_{k+1} - Z_k = \beta S_{k+1}$ shows that $2\beta\langle S_{k+1}, Y_{k+1} - Y_k\rangle \geq 0$. From (6.2.20) it follows that

$$\beta\sum_{k=0}^{\infty}\left(\|S_{k+1}\|_F^2 - \beta\|Y_{k+1} - Y_k\|_F^2\right) \leq F^0, \tag{6.2.39}$$

which implies that $S_{k+1} \longrightarrow 0$ and $Y_{k+1} - Y_k \longrightarrow 0$ as $k \longrightarrow \infty$. It follows then from (6.2.22) and (6.2.28) that $\lim_{k\to+\infty} F(X_{k+1}) + G(X_{k+1}) = F(X_*) + G(X_*)$,

6.3 TV/L1 minimization problem

In this section we consider the following regularized minimization problem

$$\min_X \|\mathcal{H}(X) - B\|_{1,1} + \mu \mathrm{TV}_2(X) \tag{6.3.1}$$

We first notice that the minimization problem (6.3.1) can be rewritten as

$$\min_X \left(\|\mathcal{H}(X) - B\|_{1,1} + \mu \sum_{i=1}^{m} \sum_{j=1}^{n} \|M_{i,j}\|_2 \right), \tag{6.3.2}$$

then, the constraint violation of the problem (6.3.1) can be written as follows

$$\begin{aligned} &\min_{X,R,M^{(n)},M^{(m)}} \left(\|R - B\|_{1,1} + \mu \sum_{i=1}^{m} \sum_{j=1}^{n} \|M_{i,j}\|_2 \right), \\ &\text{subject to} \quad D_{1,n}X = M^{(n)}, \quad D_{1,m}X = M^{(m)}, \quad R = \mathcal{H}(X). \end{aligned} \tag{6.3.3}$$

This constrained problem can be also reformulated as

$$\begin{aligned} &\min \quad F(R) + G(Y), \\ &\text{subject to} \quad DX = Y, \quad \mathcal{H}(X) = R \end{aligned} \tag{6.3.4}$$

where,

$$F(R) = \|R - B\|_{1,1}, \quad G(Y) = \mu \sum_{i=1}^{m} \sum_{j=1}^{n} \|M_{i,j}\|_2, \quad D = \begin{pmatrix} D_{1,n} \\ \\ D_{1,m} \end{pmatrix}, \quad Y = \begin{pmatrix} M^{(n)} \\ \\ M^{(m)} \end{pmatrix},$$

The problem now fits the framework of the augmented Lagrangian method [59, 85] which puts a quadratic penalty term instead of the constraint in the objective function and introducing explicit Lagrangian multipliers at each iteration into the objective function. The augmented Lagrangian function of (6.3.4) is defined as follows

$$\begin{aligned} &\mathcal{L}(X, R, Y, Z, W) = \\ &F(R) + G(Y) + \frac{\beta}{2}\|DX - Y\|_F^2 + \langle DX - Y, Z\rangle_F + \frac{\rho}{2}\|\mathcal{H}(X) - R\|_F^2 + \langle \mathcal{H}(X) - R, W\rangle_F \end{aligned} \tag{6.3.5}$$

$Z \in \mathbb{R}^{2m\times n}$ and $W \in \mathbb{R}^{m\times n}$ are the Lagrange multipliers of the linear constraint $DX = Y$ and $R = \mathcal{H}(X)$, respectively. The parameters $\beta > 0$ and $\rho > 0$ are the penalty parameters for the violation of the linear constraint.

Again, we use the ADM method to solve the nonlinear problem (6.3.1), by finding the saddle point of the Lagrangian (6.3.5). Therefore, for $k = 0, 1, ...$ we solve

$$(X_k, R_k, Y_k) = \arg\min_{X,R,Y} \mathcal{L}_{\beta,\rho}(X, R, Y, Z_k, W_k). \tag{6.3.6}$$

The Lagrange multipliers are updated by

$$\begin{aligned} Z_{k+1} &= Z_k + \beta\,(DX_k - Y_k)\,, \\ W_{k+1} &= W_k + \rho\,(\mathcal{H}(X_k) - R_k)\,. \end{aligned} \tag{6.3.7}$$

Next, we will see how to solve the problems (6.3.6), to determine the iterates X_k, Y_k and R_k

6.3.1 Solving the X-problem

Given Y and R, X_k can be obtained by solving the minimization problem

$$\min_X \frac{\beta}{2}\|DX - Y\|_F^2 + \langle DX - Y, Z_k\rangle_F + \frac{\rho}{2}\|\mathcal{H}(X) - R\|_F^2 + \langle \mathcal{H}(X) - R, W_k\rangle_F \tag{6.3.8}$$

The problem (6.3.8) is now continuously differentiable at X. Therefore, it can be solved by considering the following normal equation

$$\rho H_1^T H_1 X H_2^T H_2 + \beta D^T DX = H_1^T\,(\rho R - W_k)\,H_2 + D^T\,(\beta Y - Z_k)\,. \tag{6.3.9}$$

The linear matrix equation (6.3.9) can be rewritten in the following form

$$A_1 X A_2 + A_3 X A_4 = E_k, \tag{6.3.10}$$

where $A_1 = \rho H_1^T H_1$, $A_2 = H_2^T H_2$, $A_3 = \beta D^T D$, $A_4 = I$ and $E_k = H_1^T (\rho R - W_k) H_2$ $+D^T (\beta Y - Z_k)$.

The equation (6.3.10) is refereed to as the generalized Sylvester matrix equation.

6.3.2 Solving the R-problem

Given X, the iterate R_k can be obtained by solving the minimization problem

$$\min_R \|R - B\|_{1,1} + \frac{\rho}{2} \|\mathcal{H}(X) - R\|_F^2 + \langle \mathcal{H}(X) - R, W_k \rangle_F . \tag{6.3.11}$$

Therefore, by using the following well-known one-dimensional Shrinkage formula [75]

$$\textbf{Shrink}(y, \gamma, \delta) = \max \left\{ \left| y + \frac{\gamma}{\delta} \right| - \frac{1}{\delta}, 0 \right\} .\text{sign} \left(y + \frac{\gamma}{\delta} \right), \tag{6.3.12}$$

the minimizer of (6.3.11) is then given by

$$\max \left\{ \left| \mathcal{H}(X) - B + \frac{1}{\rho} W \right| - \frac{1}{\rho}, 0 \right\} .\text{sign} \left(\mathcal{H}(X) - B + \frac{1}{\rho} W \right). \tag{6.3.13}$$

6.3.3 Solving the Y-problem

Given X and R, we compute the iterates Y_k by solving the problem

$$\min_Y \mu \sum_{i=1}^{m} \sum_{j=1}^{n} \|M_{i,j}\|_2 + \frac{\beta}{2} \|DX - Y\|_F^2 + \langle DX - Y, Z_k \rangle_F . \tag{6.3.14}$$

This solution can be obtained by equation (6.2.11), since the minimization problem (6.3.14) is the same as that of TV/L2.

6.3.4 Convergence analysis of TV/L1 problem

In this subsection we study the convergence of Algorithm 11 used to solve the TV/L1 problem. Note that the convergence study for TV/L2 does not hold for TV/L1 problem

since in general $\beta \neq \rho$ in (6.3.5). For the problem (6.3.4), F and G are closed proper convex functions. According to [35, 89], the problem (6.3.4) is solvable, i.e., there exist R^* and Y^*, not necessarily unique that minimize (6.3.4). Let $\mathcal{W} = \Omega \times \mathcal{Y} \times \mathcal{X} \times \mathbb{R}^{2m\times n} \times \mathbb{R}^{m\times n}$, where Ω, $\mathcal{X}$ and $\mathcal{Y}$ are given closed and convex nonempty sets. The saddle-point problem is equivalent to finding $(X_*, R_*, Y_*, Z_*, W_*) \in \mathcal{W}$ such that

$$\mathcal{L}_{\beta,\rho}(X_*, R_*, Y_*, Z, W) \leq \mathcal{L}_{\beta,\rho}(X_*, R_*, Y_*, Z_*, W_*) \leq \mathcal{L}_{\beta,\rho}(X, R, Y, Z_*, W_*),$$
$$\forall\, (X, R, Y, Z, W) \in \mathcal{W}. \tag{6.3.15}$$

The properties of the relation between the saddle-points of $\mathcal{L}_{\beta,\rho}$ and the solution of (6.3.4) are stated by the following theorem from [102]

Theorem 6.3.1. *X_* is a solution of (6.3.1) if and only if there exist $(R_*, Y_*) \in \mathcal{Y} \times \mathcal{X}$ and $(Y_*, Z_*) \in \mathbb{R}^{2m\times n} \times \mathbb{R}^{m\times n}$ such that $(X_*, R_*, Y_*, Z_*, W_*)$ is a saddle-point of (6.3.15)*

The convergence of ADM for TV/L1 has been well studied in the literature in the context of vectors ; see, e.g., [102]. Our TV/L1 problem is a model with matrix variables, it is our aim to give a similar convergence results for the matrix case

Theorem 6.3.2. *Assume that $(X_*, R_*, Y_*, Z_*, W_*)$ is a saddle-point of $\mathcal{L}_{\beta,\rho}$. The sequence $(X_k, R_k, Y_k, Z_k, W_k)$ generated by Algorithm 11 satisfies*

1. $\lim_{k\to+\infty} F(R_k) + G(Y_k) = F(R_*) + G(Y_*)$,
2. $\lim_{k\to+\infty} \|DX_k - Y_k\|_F = 0$,
3. $\lim_{k\to+\infty} \|\mathcal{H}(X_k) - R_k\|_F = 0$.

Proof From the first inequality of (6.3.15) it follows that $\forall (Z, W) \in \mathbb{R}^{2m\times n} \times \mathbb{R}^{m\times n}$

$$\langle DX_* - Y_*, Z_*\rangle_F + \langle \mathcal{H}(X_*) - B - R_*, W_*\rangle_F \leq \langle DX_* - Y_*, Z\rangle_F + \langle \mathcal{H}(X_*) - B - R_*, W\rangle_F\,, \tag{6.3.16}$$

which obviously implies that

$$\begin{aligned} DX_* &= Y_*, \\ \mathcal{H}(X_*) &= R_*. \end{aligned} \tag{6.3.17}$$

Let us define the following quantities

$$\overline{Z}_k = Z_k - Z_*, \quad \overline{W}_k = W_k - W_*, \quad \overline{X}_k = X_k - X_*, \quad \overline{R}_k = R_k - R_*, \quad \overline{Y}_k = Y_k - Y_*.$$

With the relationship (6.3.17) together with (6.3.7), we can define

$$\overline{Z}_{k+1} = \overline{Z}_k + \beta\left(D\overline{X}_k - \overline{Y}_k\right) \tag{6.3.18}$$

$$\overline{W}_{k+1} = \overline{W}_k + \rho\left(\mathcal{H}(\overline{X}_k) - \overline{R}_k\right) \tag{6.3.19}$$

In order to show the convergence, it suffice to show that $\left(\beta\|\overline{Z}_k\|_F^2 + \rho\|\overline{W}_k\|_F^2\right)$ decreases at each iteration. In the following we show that

$$\left(\beta\|\overline{Z}_k\|_F^2 + \rho\|\overline{W}_k\|_F^2\right) - \left(\beta\|\overline{Z}_{k+1}\|_F^2 + \rho\|\overline{W}_{k+1}\|_F^2\right) \tag{6.3.20}$$

$$\geq \beta^2\rho\|D\overline{X}_k - \overline{Y}_k\|_F^2 + \beta\rho^2\|\mathcal{H}(\overline{X}_k) - \overline{R}_k\|_F^2. \tag{6.3.21}$$

For $(X, R, Y) = (X_k, R_k, Y_k)$ in (6.3.15) , the second equality implies

$$\left\langle D^T Z_*, X_k - X_*\right\rangle_F + \beta\left\langle D^T(Y_* - DX_*), X_k - X_*\right\rangle_F \tag{6.3.22}$$

$$+ \quad \langle W_*, -\mathcal{H}(X_k - X_*)\rangle_F + \rho\left\langle Z_* - \mathcal{H}(X_*), -\mathcal{H}(X_k - X_*)\right\rangle_F \geq 0,$$

$$F(R_k) - F(R_*) + \langle W_*, R_k - R_*\rangle_F + \rho\left\langle R_* - \mathcal{H}(X_*), R_k - R_*\right\rangle_F \geq 0, \tag{6.3.23}$$

$$G(Y_k) - G(Y_*) + \langle Z_*, Y_k - Y_* \rangle_F + \beta \langle Y_* - DX_*, Y_k - Y_* \rangle\rangle_F \geq 0. \tag{6.3.24}$$

Since (X_k, R_k, Y_k) is also a saddle-point of $\mathcal{L}_{\beta,\rho}$, for $(X, R, Y) = (X_*, R_*, Y_*)$ the second equality of (6.3.15) implies

$$\begin{aligned} & \left\langle D^T Z_k, X_* - X_k \right\rangle_F + \beta \left\langle D^T (Y_k - DX_k), X_* - X_k \right\rangle_F \\ + \quad & \langle W_k, -\mathcal{H}(X_* - X_k) \rangle_F + \rho \langle Z_k - \mathcal{H}(X_k), -\mathcal{H}(X_* - X_k) \rangle_F \geq 0, \end{aligned} \tag{6.3.25}$$

$$F(R_*) - F(R_k) + \langle W_k, R_* - R_k \rangle_F + \rho \langle R_k - \mathcal{H}(X_k), R_* - R_k) \rangle \geq 0, \tag{6.3.26}$$

$$G(Y_*) - G(Y_k) + \langle Z_k, Y_* - Y_k \rangle_F + \beta \langle Y_k - DX_k, Y_* - Y_k) \rangle \geq 0. \tag{6.3.27}$$

By addition , regrouping terms, and multiplying through by $\beta\rho$ gives

$$-\beta\rho \left\langle \overline{Z}_k, D\overline{X}_k - \overline{Y}_k \right\rangle - \beta\rho \left\langle \overline{W}_k, \mathcal{H}(\overline{X}_k) - \overline{Z}_k \right\rangle \geq \beta^2\rho \|D\overline{X}_k - \overline{Y}_k\|_F^2 + \beta\rho^2 \|\mathcal{H}(\overline{X}_k) - \overline{R}_k\|_F^2 \tag{6.3.28}$$

It is now suffice to show that

$$\begin{aligned} \left(\beta\|\overline{Z}_k\|_F^2 + \rho\|\overline{W}_k\|_F^2\right) \quad - \quad & \left(\beta\|\overline{Z}_{k+1}\|_F^2 + \rho\|\overline{W}_{k+1}\|_F^2\right) \\ \geq \quad & -\beta\rho \left\langle \overline{Z}_k, D\overline{X}_k - \overline{Y}_k \right\rangle - \beta\rho \left\langle \overline{W}_k, \mathcal{H}(\overline{X}_k) - \overline{Z}_k \right\rangle \end{aligned} \tag{6.3.29}$$

To show this, we see that (6.3.18) is equivalent to

$$\begin{aligned} \sqrt{\rho}\overline{Z}_{k+1} \quad &= \quad \sqrt{\rho}\overline{Z}_k + \beta\sqrt{\rho}\left(D\overline{X}_k - \overline{Y}_k\right) \\ \sqrt{\beta}\overline{W}_{k+1} \quad &= \quad \sqrt{\beta}\overline{W}_k + \rho\sqrt{\beta}\left(\mathcal{H}(\overline{X}_k) - \overline{R}_k\right) \end{aligned} \tag{6.3.30}$$

the result follows then from the equality

$$\begin{aligned} & \left(\beta\|\overline{Z}_k\|_F^2 + \rho\|\overline{W}_k\|_F^2\right) - \left(\beta\|\overline{Z}_{k+1}\|_F^2 + \rho\|\overline{W}_{k+1}\|_F^2\right) \\ = \quad & -2\beta\rho \left\langle \overline{Z}_k, D\overline{X}_k - \overline{Y}_k \right\rangle - 2\beta\rho \left\langle \overline{W}_k, \mathcal{H}(\overline{X}_k) - \overline{Z}_k \right\rangle - \beta^2\rho \|D\overline{X}_k - \overline{Y}_k\|_F^2 - \beta\rho^2 \|\mathcal{H}(\overline{X}_k) - \overline{R}_k\|_F^2 \end{aligned} \tag{6.3.31}$$

It follows that

$$\sum_{k=0}^{\infty} \left(\beta^2\rho\|D\overline{X}_k - \overline{Y}_k\|_F^2 + \beta\rho^2\|\mathcal{H}(\overline{X}_k) - \overline{R}_k\|_F^2\right) \leq \left(\beta\|\overline{Z}_0\|_F^2 + \rho\|\overline{W}_0\|_F^2\right), \qquad (6.3.32)$$

which implies that $D\overline{X}_k - \overline{Y}_k \longrightarrow 0$ and $\mathcal{H}(\overline{X}_k) - \overline{R}_k \longrightarrow 0$ as $k \longrightarrow \infty$.

To show $\lim_{k\to+\infty} F(R_k) + G(Y_k) = F(R_*) + G(Y_*)$, we first see that the second inequality of (6.3.15) implies

$$\begin{aligned} F(R_*) + G(Y_*) - F(R_k) - G(Y_k) &\leq \langle W_*, \mathcal{H}(X_k) - R_k\rangle_F + \langle Z_*, DX_k - Y_k\rangle_F \\ &+ \beta\|DX_k - Y_k\|_F^2 + \|\mathcal{H}(X_k) - R_k\|_F^2 \qquad (6.3.33) \end{aligned}$$

in the other hand, by addition of (6.3.25), (6.3.26) and (6.3.27) we obtain

$$\begin{aligned} F(R_k) + G(Y_k) - F(R_*) - G(Y_*) &\leq -\langle W_k, \mathcal{H}(X_k) - R_k\rangle_F - \langle Z_k, DX_k - Y_k\rangle_F \\ &- \beta\|DX_k - Y_k\|_F^2 - \|\mathcal{H}(X_k) - R_k\|_F^2, \qquad (6.3.34) \end{aligned}$$

thus we have $\lim_{k\to+\infty} F(R_k) + G(Y_k) = F(R_*) + G(Y_*)$, i.e., objective convergence.

6.4 Generalized matrix Krylov subspace for TV/L1 and TV/L2 regularizations

In this section we will see how to generalize the generalized Krylov subspace (GKS) method proposed in [78] to solve the generalized Sylvester matrix equation (6.2.17). In [78] GKS was introduced to solve Tikhonov regularization problems with a generalized regularization matrix. The method was next generalized in [76] to iteratively solve a sequence of weighted ℓ_2-norms. It is our aim to use the fashion of the GKS method to iteratively solve the sequence of generalized Sylvester matrix equation (6.2.17). Let us

first introduce the following linear matrix operator

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{m\times n} &\rightarrow \mathbb{R}^{m\times n} \\ \mathcal{A}(X) &:= A_1 X A_2 + A_3 X A_4. \end{aligned}$$

the problem (6.2.17) can be then expressed as follows

$$\mathcal{A}(X) = E_k, \quad k = 0, 1, ... \tag{6.4.1}$$

We start with the solution X_1 of the following linear matrix equation

$$\mathcal{A}(X) = E_0 \tag{6.4.2}$$

We search for an approximation of the solution by solving the following minimization problem,

$$\min_X \ \|\mathcal{A}(X) - E_0\|_F \tag{6.4.3}$$

Let X_0 be an initial guess of X_1 and $P_0 = \mathcal{A}(X) - E_0$ the corresponding residual. We use the modified global Arnoldi algorithm [66] to construct an F-orthonormal basis $\mathcal{V}_m = [V_1, V_2, ..., V_m]$ of the following matrix Krylov subspace

$$\mathcal{K}_m(\mathcal{A}, P_0) = \text{span}\left\{P_0, \mathcal{A}(P_0), ..., \mathcal{A}^{m-1}(P_0)\right\}. \tag{6.4.4}$$

This gives the following relation

$$\mathcal{A}(\mathcal{V}_m) = \mathcal{V}_{m+1}(H_m \otimes I_n), \tag{6.4.5}$$

where $H_m \in \mathbb{R}^{(m+1)\times m}$ is an upper Hessenberg matrix. We search for an approximated solution X_1^m of X_1 belonging to $X_0 + \mathcal{K}_m(\mathcal{A}, P_0)$. This shows that X_1^m can be obtained as follows

$$X_1^m = X_0 + \mathcal{V}_m(y_m \otimes I_n), \tag{6.4.6}$$

where y_m is the solution of the following reduced minimization problem

$$\min_{y\in\mathbb{R}^m} \|H_m y - \|P_0\|_F e_1\|, \tag{6.4.7}$$

where e_1 denotes the first unit vector of $\mathbb{R}^{m+1}$.

Now we turn to the solutions of

$$\mathcal{A}(X) = E_k, \quad k = 1, 2, ... \tag{6.4.8}$$

For example, in the beginning of solving $\mathcal{A}(X) = E_1$, we reuse the F-orthonormal vectors $\mathcal{V}_m$ and we expand it to $\mathcal{V}_{m+1} = [\mathcal{V}_m, V_{\text{new}}]$, where V_{new} is obtained normalizing the residual as follows

$$V_{\text{new}} = \frac{P_1}{\|P_1\|_F}, \quad P_1 = \mathcal{A}(X_1) - E_1. \tag{6.4.9}$$

We can then continue with $\mathcal{A}(X) = E_k$, $k = 2, 3, ...$ in a similar manner. Thus, at each iteration we generate the following new vector that has to be added to the generalized matrix Krylov subspace already generated to solve all the previous matrix equation,

$$V_{\text{new}} = \frac{P_k}{\|P_k\|_F}, \quad P_k = \mathcal{A}(X_k) - E_k. \tag{6.4.10}$$

The idea of reusing these vectors to solve the next matrix equation, generates matrix subspaces refereed to as generalized matrix Krylov subspaces of increasing dimension [12]. Note that at each iteration, the residual P_k is orthogonal to $\mathcal{V}_k$, since it is parallel to the gradient of the function (6.2.15) evaluated at X_k. Let $\mathcal{V}_k$ be the F-orthonormal basis of the generalized matrix Krylov subspaces at iteration k. When solving $\mathcal{A}(X) = E_k$, given X_k and the corresponding residual P_k, in order to minimize the residual in the generalized matrix Krylov subspaces spanned by $\mathcal{V}_k$, we need to solve the following minimization

problem

$$\min_{X \in \mathrm{span}(\mathcal{V}_k)} \|P_k - \mathcal{A}(X)\|_F, \tag{6.4.11}$$

The approximate solution of (6.4.11) is then given by $X_{k+1} = \mathcal{V}_k (y \otimes I_n)$. By means of the Kronecker product, we can recast (6.4.11) to a vector least-squares problem. Hence, replacing the expression of X_{k+1} into (6.4.11) yields the following minimization problem

$$\min_{y_k} \|P_k - [\mathcal{A}(V_1), ..., \mathcal{A}(V_k), \mathcal{A}(V_{\mathrm{new}})] (y \otimes I_n)\|_F, \tag{6.4.12}$$

The problem (6.4.12) can be solved by the updated version of the global QR decomposition [14]. To use the global QR decomposition, we first need to define the $\diamond$ product. Let $A = [A_1, A_2, ..., A_p]$ and $B = [B_1, B_2, ..., B_\ell]$ be matrices of dimension $n \times ps$ and $n \times \ell s$, respectively, where A_i and B_j $(i = 1, ..., p; j = 1, ..., \ell)$ are $n \times s$ matrices. Then the $p \times \ell$ matrix $A^T \diamond B$ is defined by

$$A^T \diamond B = \begin{bmatrix} \langle A_1, B_1 \rangle_F & \langle A_1, B_2 \rangle_F & \cdots & \langle A_1, B_\ell \rangle_F \\ \langle A_2, B_1 \rangle_F & \langle A_2, B_2 \rangle_F & \cdots & \langle A_1, B_\ell \rangle_F \\ \vdots & \vdots & & \vdots \\ \langle A_p, B_1 \rangle_F & \langle A_p, B_2 \rangle_F & \cdots & \langle A_p, B_\ell \rangle_F \end{bmatrix}. \tag{6.4.13}$$

Let $Q_{\mathcal{A}} (R_{\mathcal{A}} \otimes I_n)$ be the global QR of $[\mathcal{A}(V_1), \mathcal{A}(V_2), ..., \mathcal{A}(V_k)]$, where $Q_{\mathcal{A}} = [Q_1, ..., Q_k]$ is an $m \times kn$ F-orthonormal matrix satisfying $Q_{\mathcal{A}}^T \diamond Q_{\mathcal{A}} = I_k$ and $R_{\mathcal{A}}$ is an upper triangular $k \times k$ matrix. The global QR decomposition of $[\mathcal{A}(V_1), ..., \mathcal{A}(V_k), \mathcal{A}(V_{\mathrm{new}})]$ is defined as follows

$$[\mathcal{A}(V_1), ..., \mathcal{A}(V_k), \mathcal{A}(V_{\mathrm{new}})] = [Q_{\mathcal{A}}, Q_{\mathrm{new}}] \left(\begin{bmatrix} R_{\mathcal{A}} & r_{\mathcal{A}} \\ 0 & r_a \end{bmatrix} \otimes I_n \right), \tag{6.4.14}$$

where Q_{new}, $r_{\mathcal{A}}$ and r_a are updated as follows

$$\begin{aligned} r_{\mathcal{A}} &= Q_{\mathcal{A}}^T \diamond \mathcal{A}(V_{\text{new}}), \quad Q = \mathcal{A}(V_{\text{new}}) - Q_{\mathcal{A}} \left(r_{\mathcal{A}} \otimes I_n\right) \\ r_a &= \|Q\|_F, \qquad\qquad\qquad Q_{\text{new}} = Q/r_a. \end{aligned} \tag{6.4.15}$$

Algorithm 10 TV/L2 for (6.1.15)

Inputs : H_1, H_2, C, B, ε

Initialization : $X_0 = B$, $Y_0 = DX_0$, $Z_0 = 0$

Parameters : μ, β

1. Generate matrix Krylov subspace $\mathcal{V}_m$ using modified global Arnoldi's process. Set $X_1 = X_1^m$, where X_1^m is obtained by (6.4.6)
2. **For** $k = 1, ...$ until convergence, **do**
3. Update Y_k by (6.2.11) and Z_k by (6.2.6)
4. Calculate $P_k = \mathcal{A}\left(X_k\right) - E_k$, where $E_k = \mathcal{H}^T(B) + D^T\left(\beta Y_k - Z_k\right)$
5. Calculate $V_{\text{new}} = \frac{P_k}{\|P_k\|_F}$ and save $\mathcal{V}_{k+1} = [\mathcal{V}_k, V_{\text{new}}]$
6. Update X_{k+1} by solving $\min\limits_{X \in \text{span}(\mathcal{V}_{k+1})} \|P_k - \mathcal{A}\left(X\right)\|_F$ with the updated global QR decomposition
7. End the iteration if $\|X_{k+1} - X_k\|_F \,/\, \|X_k\|_F < \varepsilon$

6.5 Numerical results

This section provides some numerical results to show the performance of Algorithms TV/L1 and TV/L2 when applied to the restoration of blurred and noisy images. The

Algorithm 11 TV/L1 for (6.1.16)

Inputs : H_1, H_2, C, B, ε

Initialization : $R_0 = \mathcal{A}(X_0) - B$, $Y_0 = DX_0$, $Z_0 = 0$, $W_0 = 0$

Parameters : μ, β, ρ

1. Generate matrix Krylov subspace $\mathcal{V}_m$ using modified global Arnoldi's process. Set $X_1 = X_1^m$, where X_1^m is obtained by (6.4.6)
2. **For** $k =, 1, ...$ until convergence, **do**
3. Update R_k by (6.3.13) and Update Y_k by (6.2.11)
4. Update Z_k and W_k by (6.3.7)
5. Calculate $P_k = \mathcal{A}(X_k) - E_k$, where $E_k = \mathcal{H}^T(\rho R_k - W_k) + D^T(\beta Y_k - Z_k)$
6. Calculate $V_{\text{new}} = \frac{P_k}{\|P_k\|_F}$ and save $\mathcal{V}_{k+1} = [\mathcal{V}_k, V_{\text{new}}]$
7. Update X_{k+1} by solving $\min\limits_{X \in \text{span}(\mathcal{V}_{k+1})} \|P_k - \mathcal{A}(X)\|_F$ with the updated global QR decomposition
8. End the iteration if $\|X_{k+1} - X_k\|_F / \|X_k\|_F < \varepsilon$

first example applies TV/L1 to the restoration of blurred image contaminated Gaussian blur salt-and-pepper noise while the second example apply the TV/L1 model when also a color image is contaminated by Gaussian blur salt-and-pepper noise. The third example discusses TV/L2 when applied to the restoration of an image that have been contaminated by Gaussian blur and by additive zero-mean white Gaussian noise. All computations were carried out using the MATLAB environment on an Pentium(R) Dual-Core CPU T4200 computer with 3 GB of RAM. The computations were done with approximately 15 decimal digits of relative accuracy. To determine the effectiveness of our solution methods, we evaluate the Signal-to-Noise Ratio (SNR) defined by

$$\text{SNR}(X_k) = 10\log_{10}\frac{\|\widehat{X} - E(\widehat{X})\|_F^2}{\|X_k - \widehat{X}\|_F^2}$$

where $E(\widehat{X})$ denotes the mean gray-level of the uncontaminated image $\widehat{X}$. The parameters are chosen empirically to yield the best reconstruction. In all the examples we generate the matrix Krylov subspace $\mathcal{V}_1$ using only one step of the modified global Arnoldi's process.

Example 1

In this example the original image is the gray-scale `mrin6.png` image of dimension 256×256 from Matlab and it is shown in Figure 6.1. The blurring matrix H is given by $H = H_1 \otimes H_2 \in \mathbb{R}^{256^2 \times 256^2}$, where $H_1 = H_2 = [h_{ij}]$ and $[h_{ij}]$ is the Toeplitz matrix of dimension 256×256 given by

$$h_{ij} = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(i-j)^2}{2\sigma^2}\right), |i-j| \leq r, \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

The blurring matrix H models a blur arising in connection with the degradation of digital images by atmospheric turbulence blur. We let $\sigma = 1$ and $r = 4$. The blurred and noisy

image of Figure 6.2 has been built by the product $H_2\hat{X}H_1^T$ and by adding salt-and-pepper noise of different intensity. The recovery of the image via $TV_1/L1$ and $TV_2/L1$ models is terminated as soon as $\|X_{k+1} - X_k\|_F / \|X_k\|_F < 10^{-3}$. Table 6.1 report results of the performances of the TV models for different percentages of pixels corrupted by salt-and-pepper noise. In Figures 6.3-6.4 we show the resorted images obtained applying TV/L1 algorithm for 30% noise level.

	Parameters			TV_1			TV_2		
Noise %	μ	β	ρ	Iter	SNR	time	Iter	SNR	time
10	0.05	50	5	56	23.55	10.23	141	22.64	42.55
20	0.1	50	5	51	21.38	8.69	106	20.16	27.16
30	0.2	50	5	48	19.21	7.68	87	17.66	19.73

TABLE 6.1 – Comparison of $TV_1/L1$ and $TV_2/L1$ models for the restoration of `mrin6.png` test image corrupted by Gaussian blur and different salt-and-pepper noise.

6.5.1 Example 2

This example illustrates the performance of TV/L1 algorithm when applied to the restoration of 3-channel RGB color images that have been contaminated by blur and salt and peppers noise. The corrupted image is stored in a block vector B with three columns. The desired (and assumed unavailable) image is stored in the block vector $\widehat{X}$ with three columns. The blur-contaminated, and noisy image associated with $\widehat{X}$, is stored in the block vector B.

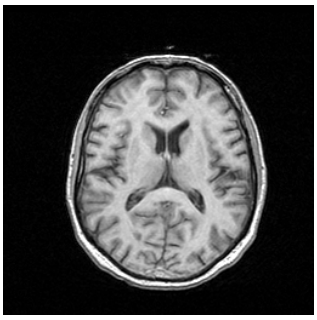


FIGURE 6.1 – Original image

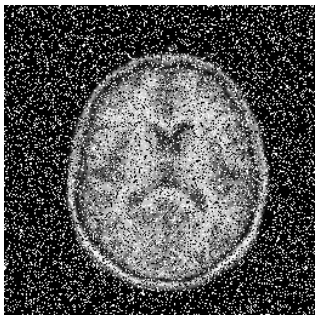


FIGURE 6.2 – Corrupted

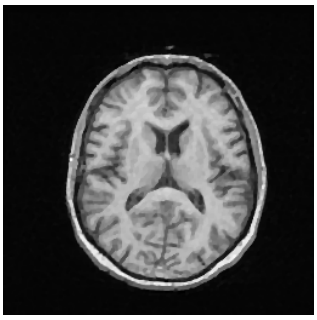


FIGURE 6.3 – TV_1 (SNR=19.21)

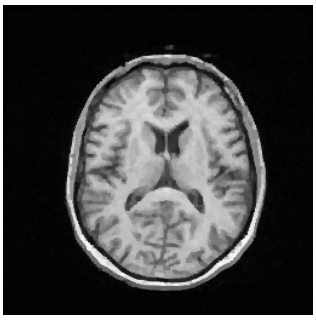


FIGURE 6.4 – TV_2 (SNR=17.66)

We consider the within-channel blurring only. Hence the blurring matrix H_1 in (6.1.1) is the 3×3 identity matrix. The blurring matrix H_2 in (6.1.1), which describes the blurring within each channel, models Gaussian blur and is determined with the MATLAB function `blur` from [61]. This function has two parameters, the half-bandwidth of the Toeplitz blocks r and the variance σ of the Gaussian PSF. For this example we let $\sigma = 1$ and $r = 4$. The original (unknown) RGB image $\widehat{X} \in 256 \times 256 \times 3$ is the `papav256` image from

MATLAB. It is shown in Figure 6.5. The associated blurred and noisy image B with 30% noise level is shown in Figure 6.6. Given the contaminated image B, we would like to recover an approximation of the original image $\widehat{X}$. The recovery of the image via $TV_1/L1$ and $TV_2/L1$ models is terminated as soon as $\|X_{k+1} - X_k\|_F / \|X_k\|_F < 10^{-2}$. Table 6.2 compares the results obtained by $TV_1/L1$ and $TV_2/L1$ models.

The restorations obtained with $TV_1/L1$ and $TV_2/L1$ for noise level 30% are shown in Figure 6.7 and the Figure 6.8, respectively.

	Parameters			TV_1			TV_2		
Noise %	μ	β	ρ	Iter	SNR	time	Iter	SNR	time
10	0.1	80	5	13	24.66	9.01	14	24.32	9.73
20	0.125	80	5	17	23.00	12.64	17	22.71	12.36
30	0.125	80	5	19	20.90	13.35	19	21.13	13.89

TABLE 6.2 – Comparison of $TV_1/L1$ and $TV_2/L1$ models for the restoration of `papav256.png` test colour image corrupted by Gaussian blur and different salt-and-pepper noise.

6.5.2 Example 3

In this example we present the experimental results recovered by Algorithm 10 for the reconstruction of a cross-channel blurred image. We consider the same original RGB image and the same within-channel blurring matrix H_1, as in Example 2, with the same parameters. The cross-channel blurring is determined by a matrix H_2. In our example we

FIGURE 6.5 – Original image

FIGURE 6.6 – Corrupted

FIGURE 6.7 – TV_1 (SNR=20.90)

FIGURE 6.8 – TV_2 (SNR=21.13)

let H_2 to be

$$H_2 = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.25 & 0.5 & 0.25 \\ 0.15 & 0.1 & 0.75 \end{bmatrix}.$$

This matrix is obtained from [62]. The cross-channel blurred image without noise is represented by $H_1 \widehat{X} H_2^T$ and it is shown in Figure (6.9) . The associated blurred and noisy image B with 30% noise level is shown in Figure (6.10). The cross-channel blurred and noisy image has been reconstructed using Algorithm 10 as soon as $\|X_{k+1} - X_k\|_F / \|X_k\|_F < 10^{-2}$. The restored images obtained with TV/L1 models are shown in Figures (6.12)-(6.11).

6.5.3 Example 4

In this example we consider the restoration of the gray-scale `mrin6.png` image degraded by the same blurring matrices H_1 and H_2 defined in Example 1 with $\sigma = 2$ and $r = 4$, and by additive zero-mean white Gaussian noise with different different noise levels. This noise level is defined as follows $\nu = \frac{||E||_F}{||\widehat{B}||_F}$, where E denotes the block vector that represents the noise in B, i.e., $B := \widehat{B} + E$, and $\widehat{B}$ is the noise-free image associated with original image $\widehat{X}$. For this kind of noise, we consider the $TV_1/L2$ and $TV_2/L2$ models. The recovery of the image via $TV_1/L1$ and $TV_2/L1$ models is terminated as soon as $\|X_{k+1} - X_k\|_F / \|X_k\|_F < 10^{-3}$. In Table 6.3, we compare the results obtained by $TV_1/L2$ and $TV_2/L2$ for different noise levels. Figure 6.14 shows the image degraded by 0.01 noise level. Figure 6.15 and Figure 6.16 show the restored images obtained by $TV_1/L2$ and $TV_2/L2$, respectively.

FIGURE 6.9 – Blurred image

FIGURE 6.10 – Blurred and noisy image

FIGURE 6.11 – TV_1 (SNR=19.50)

FIGURE 6.12 – TV_2 (SNR=19.90)

	Parameters		TV_1			TV_2		
Noise %	μ	β	Iter	SNR	time	Iter	SNR	time
0.001	0.0001	0.1	53	18.32	9.30	52	18.32	10.10
0.01	0.001	30	20	15.70	2.65	21	15.60	2.60

TABLE 6.3 – Comparison of $TV_1/L2$ and $TV_2/L2$

models for the restoration of `imrin6.png` test image corrupted by Gaussian blur and different white Gaussian noise level.

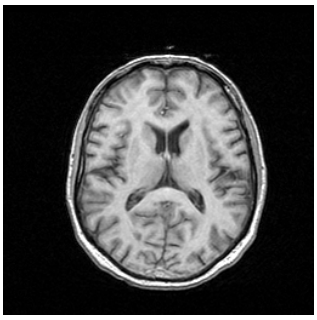


FIGURE 6.13 – Original image

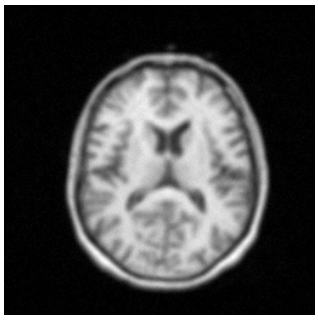


FIGURE 6.14 – Corrupted

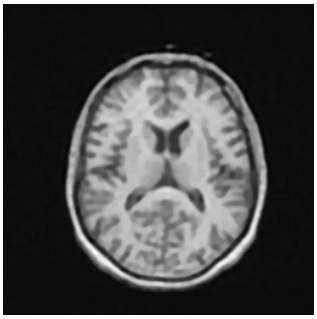


FIGURE 6.15 – TV_1 (SNR=15.70)

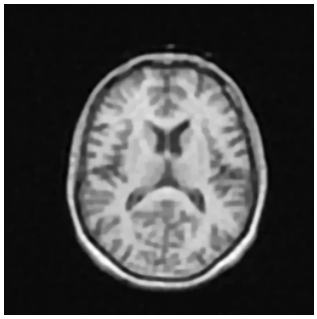


FIGURE 6.16 – TV_2 (SNR=15.60)

6.5.4 Example 5

In this example, we consider the Fredholm integral equation

$$\int \int_{\Omega} K(x, y, s, t) f(s, t) ds dt = g(x, y), \qquad (x, y) \in \Omega, \tag{6.5.1}$$

where $\Omega = [-6, 6] \times [-6, 6]$. Its kernel, solution, and right-hand side are given by

$$K(x, y, s, t) = k_1(x, s) k_1(y, t), \quad (x, y) \in \Omega, \quad (s, t) \in \Omega,$$

$$f(x, y) = f_1(x) f_1(y),$$

$$g(x,y) = g_1(x)g_1(y),$$

where

$$f_1(s) := \begin{cases} 1+\cos(\frac{\pi}{3}s), & |s| \leq \frac{\pi}{3}, \\ 0, & \text{otherwise.} \end{cases}$$

$$k_1(s,x) := f_1(s-x)$$

$$g_1(s) := (6-|s|)\left(1+\frac{1}{2}\cos\left(\frac{\pi}{3}s\right)\right) + \frac{9}{2\pi}\sin\left(\frac{\pi}{3}|s|\right).$$

We use the code `phillips` from Regularization Tools [61] to discretize (6.5.1) by a Galerkin method with orthonormal box functions as test and trial functions to obtain H_1 and H_2 of size 500. From the output of the code `phillips` we determine a scaled approximation $\widehat{X} \in \mathbb{R}^{500\times 500}$ of the exact solution $f(x,y)$. Figure 6.17 displays this exact solution. To determine the effectiveness of our approach, we evaluate the relative error

$$\text{Re} = \frac{||\widehat{X} - X_k||_F}{||\widehat{X}||_F}$$

of the computed approximate solution X_k obtained with Algorithm 10. Table 6.4 shows the relative error in approximate solutions determined by Algorithm 10 for different noise levels, as well as the number of iterations required to satisfy $\|X_{k+1} - X_k\|_F / \|X_k\|_F < 10^{-3}$. Figure 6.18 displays the computed approximate solution obtained when the noise level is 0.1.

	Parameters		TV_1			TV_2		
Noise %	μ	β	Iter	Re	time	Iter	Re	time
0.001	0.0001	0.1	12	4.01×10^{-2}	9.05	9	4.71×10^{-2}	6.52
0.01	0.001	30	13	3.99×10^{-2}	9.63	13	3.98×10^{-2}	9.66
0.1	0.1	40	15	4.07×10^{-2}	10.94	15	4.07×10^{-2}	11.38

TABLE 6.4 – Comparison of $TV_1/L2$ and $TV_2/L2$

models for the solution of (6.5.1) with different white Gaussian noise level.

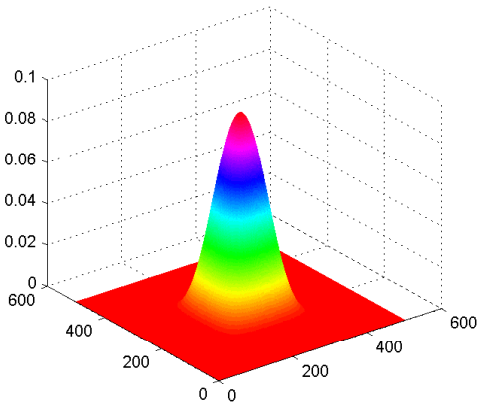


FIGURE 6.17 – True object

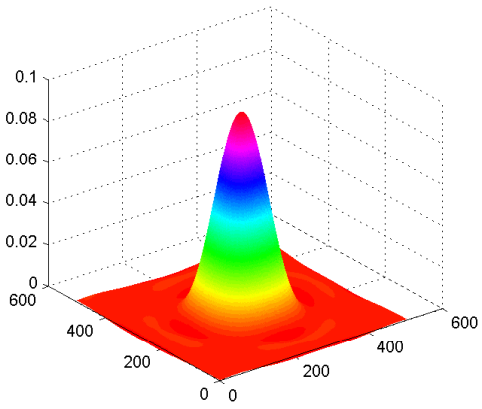


FIGURE 6.18 – Approximate solution

Chapitre 7

Conclusion

In this chapter, we provide a brief review of the main results described in this thesis.

In this thesis, we analyzed different algorithms to the solution of huge ill-posed linear problems coming from many applications. The obtained numerical results show the effectiveness of the proposed algorithms.

In Chapter 2, a new method for the inexpensive computation of a suitable value of the regularization parameter and an associated regularized solution of large linear discrete ill-posed problems is described. The method is based on global Lanczos tridiagonalization and applies Gauss-type quadrature rules to estimate pertinent quantities. Computed examples illustrate the effectiveness of the method.

In Chapter 3, an iterative scheme based on the global Golub–Kahan bidiagonalization method for the approximate solution of the Tikhonov minimization problem is presented when the matrix coefficients matrix has a Kronecker structure. The method exploits the relation between global Golub–Kahan bidiagonalization and Gauss-type quadrature to inexpensively determine the regularization parameter so that the discrepancy principle is satisfied. The Kronecker structure makes it possible to replace matrix-vector product eva-

luations in the scheme [26] by matrix-matrix product computations. The latter products execute more efficiently than the former on many computers.

In chapter 4, we have illustrated how block Lanczos algorithm, block Gauss quadrature, block anti-Gauss quadrature and matrix functional can be applied to inexpensively compute the regularization parameter for Tikhonov regularization, under the assumption that a constraint on the norm of the solution is known. The numerical experiments performed so far show that our approach is advantageous for large ill-posed linear problems. Indeed, it determines a monotonically decreasing sequence of values of the regularization parameter by carrying a few steps of block Lanczos algorithm.

Chapter 5 discusses four approaches to the solution of linear discrete ill-posed problems with multiple right-hand sides. GKB method is clearly the least attractive of the algorithms considered. The relative merits of the other algorithms depends on how accurately the noise level is known, whether the noise-contamination of all data vectors $b^{(i)}$, $i = 1, \ldots, k$, correspond to about he same noise level, and on the computer architecture.

In Chapter 6, we have introduced new algorithms to efficiently solve the TV/L1 and TV/L2 minimization problems. The proposed algorithms are based on orthogonal projections onto generalized Krylov subspaces of increasing dimensions. We tested our algorithms to the restoration of blurred and noisy grayscale and color images. The obtained numerical results indicates that subspaces of quite small dimensions are sufficient for the determination of high quality restorations.

Bibliographie

[1] G. F. Ahmad, D. H. Brooks, and R. S. Macleod, *An admissible solution approach to inverse electrocardiography*, Ann. Biomed. Eng., 26 (1998) 278–292.

[2] H. Andrews and B. Hunt, *Digital Image Restoration*, Prentice-Hall, Engelwood Cliffs, 1977.

[3] M. L. Baart, *The use of auto-correlation for pseudo-rank determination in noisy ill-conditioned least-squares problems*, IMA J. Numer. Anal., 2 (1982), 241–247. .

[4] James Baglama, *Dealing with linear dependence during the iterations of the restarted block Lanczos methods*, Numerical Algorithms, 25 (2000) 23–36.

[5] P. Brianzi, F. Di Bendetto, and C. Estatico *Improvement of space-invariant image deblurring by preconditioned Landweber iterations* SIAM J. Sci. Comput., 30 (2008) 1430–1458.

[6] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, IOP Publishing, London, 1998.

[7] R. Berry and J. Burnell, *The Handbook of Astronomical Image Processing,* Willmann-Bell Inc., Richmond, VA, 2000.

[8] M. Bertero and P. Boccacci *Introduction to Inverse Problems in Imaging*, IOP Publishing Ltd., London, 1998.

[9] A. H. Bentbib, M. El Guide, K. Jbilou, and L. Reichel, *A global Lanczos method for image restoration*, J. Comput. Appl. Math., 300 (2016), 233–244.

[10] A. H. Bentbib, M. El Guide, K. Jbilou, and L. Reichel, *Global Golub–Kahan bidiagonalization applied to large discrete ill-posed problems*, J. Comput. Appl. Math., 322 (2017) 46–56.

[11] ÅA. Björck, *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations*, BIT 18 (1988) 659–670.

[12] A. Bouhamidi and K. Jbilou, *A note on the numerical approximate solution for generalized Sylvester Matrix equations*, Appl. Math. Comput., 206(2) (2008) 687–694.

[13] A. Bouhamidi, K. Jbilou, L. Reichel, H. Sadok, *An extrapolated TSVD method for linear discrete ill-posed problems with Kronecker structure*, Linear Algebra Its Appl., 434 (2011) 1677-1688.

[14] R. Bouyouli, K. Jbilou, R. Sadaka and H. Sadok, Convergence properties of some block Krylov subspace methods for multiple linear systems. J. Comput. Appl. Math., 196 (2006) 498–511.

[15] M. R. Banham and A. K. Katsaggelos, *Digital Image Restoration*, IEEE Signal Processing Magazine, 14(2) (1997) 24-41.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. *Distributed optimization and statistical learning via the alternating direction method of multipliers.* Foundations and Trends in Machine Learning, 3(1) (2011) 1–122.

[17] C. Brezinski, G. Rodriguez, and S. Seatzu, *Error estimates for the regularization of least squares problems*, Numer. Algorithms, 51 (2009) 61–76.

[18] B. Chalmond, *Modeling and Inverse Problems in Image Analysis*, Springer, New York, 2003.

[19] A. Chambolle, *An algorithm for total variation minimization and applications*, Journal of Mathematical Imaging and Vision, 20 (2004) 89–97.

[20] T. F. Chan, S. Esedoglu, F. Park, and A. Yip, *Recent developments in total variation image restoration*, CAM Report 05-01, Department of Mathematics, UCLA, 2004.

[21] D. Calvetti, G. H. Golub, and L. Reichel, *Estimation of the L-curve via Lanczos bidiagonalization*, BIT, 39 (1999) 603–619.

[22] D. Calvetti, P. C. Hansen, and L. Reichel, *L-curve curvature bounds via Lanczos bidiagonalization*, Electron. Trans. Numer. Anal., 14 (2002) 134–149.

[23] A. Chambolle and P. L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997) 167–188.

[24] D. Calvetti, B. Lewis, L. Reichel, and F. Sgallari, *Tikhonov regularization with nonnegativity constraint*, Electron. Trans. Numer. Anal., 18 (2004) 153–173.

[25] D. Calvetti and L. Reichel, *Tikhonov regularization with a solution constraint*, SIAM J. Sci. Comput, 26 (2004) 224–239.

[26] D. Calvetti and L. Reichel, *Tikhonov regularization of large linear problems*, BIT, 43 (2003) 263–283.

[27] T. F. Chan and W. L. Wan, *Analysis of projection methods for solving linear systems with multiple right-hand sides*, SIAM J. Stat. Comput., 18 (1997) 1698–1721.

[28] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Numerical Linear Algebra for High-Performance Computers*, SIAM, Philadelphia, 1998.

[29] D. Mackenzie, *Novel Imaging Systems Rely on Focus-Free Optics*, SIAM News, 36(6) 2003.

[30] A. El Guennouni, K. Jbilou, and H. Sadok, *A block version of BiCGSTAB for linear systems with multiple right-hand sides*, Electron. Trans. Numer. Anal., 16 (2003) 129–142.

[31] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.

[32] L. Eldèn, *Algorithms for the regularization of ill-conditioned least squares problems*, BIT 17 (1977) 134–145.

[33] L. Fox and E. T. Goodwin, *The numerical solution of non-singular linear integral equations*, Philos. Trans. R. Soc. Lond. Ser. A, Math. Phys. Eng. Sci., 245(902) (1953) 501–534.

[34] C. Fenu, D. Martin, L. Reichel, and G. Rodriguez, *Block Gauss and anti-Gauss quadrature with application to networks*, SIAM J. Matrix Anal. Appl., 34 (2013) 1655–1684.

[35] F. Facchinei and J.S. Pang, *Finite-dimensional variational inequalities and complementarity problems*, Springer Series in Operations Research, Springer-Verlag, Ber-

lin, 2003.

[36] C. Fenu, L. Reichel, and G. Rodriguez, *GCV for Tikhonov regularization via global Golub–Kahan decomposition*, Numer. Linear Algebra Appl., 23 (2016) 467–484.

[37] T. L. Faber, N. Raghunath, D. Tudorascu, and J. R. Votaw *Motion correction of PET brain images through deconvolution : I. Theoretical development and analysis in software simulations* Phys. Med. Biol., 54(3) (2009) 797–811.

[38] W. Gautschi, *The interplay between classical analysis and (numerical) linear algebra – a tribute to Gene H. Golub*, Electron. Trans. Numer. Anal., 13 (2002) 113–147.

[39] W. Gautschi, *Orthogonal Polynomials : Approximation and Computation*, Oxford University Press, Oxford, 2004.

[40] K. Gallivan, M. Heath, E. Ng, B. Peyton, R. Plemmons, J. Ortega, C. Romine, A. Sameh, and R. Voigt, *Parallel Algorithms for Matrix Computations*, SIAM, Philadelphia, 1990.

[41] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*. Springer Verlag, 2008

[42] S. Gazzola, P. Novati, and M. R. Russo, *On Krylov projection methods and Tikhonov regularization*, Electron. Trans. Numer. Anal., 44 (2015) 83–123

[43] T. Goldstein and S. Osher, *The split Bregman L1 regularized problems*, SIAM J. Imaging Sci., 2 (2009) 323-343.

[44] N. P. Galatsanos, A. K. Katsaggelos, R. T. Chin, and A. D. Hillary, *Least squares restoration of multichannel images*, IEEE Trans. Signal Proc., 39 (1991) 2222–2236.

[45] G. H. Golub, F. T. Luk, and M. L. Overton, *A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix*, ACM Trans. Math. Software, 7 (1981) 149-169.

[46] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical Analysis 1993, D. F. Griffiths and G. A. Watson, eds., Longman, Essex, (1994) 105–156.

[47] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, 2010.

[48] S. Gazzola, E. Onunwor, L. Reichel, and G. Rodriguez, *On the Lanczos and Golub–Kahan reduction methods applied to discrete ill-posed problems*, Numer. Linear Algebra Appl., 23 (2016) 187–204.

[49] C. W. Groetsch, *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind*, Pitman, Boston, 1984.

[50] D. Geman and G. Reynolds, *Constrained restoration and the recovery of discontinuities*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (1992) 367–383.

[51] A. EL Guennouni, K. Jbilou, H. Sadok, *The block Lanczos method for linear systems with multiple right-hand sides*, Appl. Numer. Math., 23(51) (2004) 243–256.

[52] A. EL Guennouni, K. Jbilon, H. Sadok, *A block BiCGSTAB algorithm for multiple linear systems*, Electron. Trans. Numer. Anal. 16 (2003) 129–142.

[53] D. Gabay and B. Mercier,*A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976)

17–40.

[54] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Transactions on Image Processing, 4 (1995) 932–94.

[55] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd. Ed.)*. The Johns Hopkins University Press, Baltimore, Maryland, 1996.

[56] G. H. Golub and U. von Matt, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991) 561–580.

[57] M. Hanke, *On Lanczos based methods for the regularization of discrete ill-posed problems BIT*, 41(5) (2001) 1008–1018.

[58] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.

[59] M. R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, 4 303–320, and in Computing Methods in Optimization Problems, 2 (Eds L.A. Zadeh, L.W. Neustadt, and A.V. Balakrishnan), Academic Press, New York, 1969.

[60] P. Hamler, T. Persons, and R. J. Plemmons, *3D Iterative Restoration of Tomosynthetic Images*, OSA Trends in Optics and Photonies, Integrated Computational Imaging Systems, OSA Technical Digest. Washington, D.C., 2002.

[61] P. C. Hansen, *Regularization tools version 4.0 for MATLAB 7.3*, Numer. Algorithms, 46 (2007) 189–194.

[62] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images : Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.

[63] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.

[64] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Engelwood Cliffs, 1989.

[65] B. He, L. Liao, D. Han and H. Yang, *A new inexact alternating directions method for monotone variational inequalities*, Math. Program., 92(1) (2002) 103-118.

[66] K. Jbilou, A. Messaoudi and H. Sadok, *Global FOM and GMRES algorithms for matrix equations*, Appl. Numer. Math, 31 (1999) 49–63.

[67] K. Jbilou, H. Sadok, and A. Tinzefte, *Oblique projection methods for linear systems with multiple right-hand sides*, Electron. Trans. Numer. Anal., 20 (2005) 119–138.

[68] J. Van, *Bringing Fuzzy Field into Focus : Image Help for the Hubble. Chicago Tribune*, 1991.

[69] S. Kindermann, *Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems*, Electron. Trans. Numer. Anal., 38 (2011) 233–257.

[70] S. Kindermann, *Discretization independent convergence rates for noise level-free parameter choice rules for the regularization of ill-conditioned problems*, Electron. Trans. Numer. Anal., 40 (2013) 58–81.

[71] J. Kamm and J. G. Nagy, *Kronecker product and SVD approximations in image restoration*, Linear Algebra Appl., 284 (1998) 177–192.

[72] J. Kamm and J. G. Nagy, *Kronecker product approximations for restoration image with reflexive boundary conditions*, SIAM J. Matrix Anal. Appl., 25 (2004) 829–841.

[73] S. Kindermann, *Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems*, Electron. Trans. Numer. Anal., 38 (2011) 233–257.

[74] D.P. Laurie, *Anti-Gaussian quadrature formulas*, Math. Comp. 65 (1996) 739-747.

[75] C. Li, *An Efficient Algorithm For Total Variation Regularization with Applications to the Single Pixel Camera and Compressive Sensing*, Ph.D. thesis, Rice University, 2009, available at http ://www.caam.rice.edu/ ?optimization/L1/TVAL3/tval3 thesis.pdf

[76] A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, *A generalized Krylov subspace method for $\ell_p - \ell_q$ minimization.* SIAM J. Sci. Comput. 37(5) (2015) 30–50.

[77] F. Li, M. K. Ng, and R. J. Plemmons, *Coupled segmentation and denoising/deblurring for hyperspectral material identification*, Numer. Linear Algebra Appl., 19 (2012) 15–17.

[78] J. Lampe, L. Reichel, and H. Voss, *Large-scale Tikhonov regularization via reduction by orthogonal projection*, Linear Algebra Appl., 436 (2012) 2845–2865.

[79] G. Lopez Lagomasino, L. Reichel, and L. Wunderlich, *Matrices, moments, and rational quadrature*, Linear Algebra Appl., 429 (2008) 2540–2554.

[80] S. R. McNown and B. R. Hunt *Approximate shift-invariance by warping shift-variant systems* In R. J. Hanisch and R. L. White, editors, The Restoration of HST Images and Spectra II, (1994) 181-187.

[81] J. Meng, P. Y. Zhu, and H.-B. Li, *A block GCROT(m,k) method for linear systems with multiple right-hand sides*, J. Comput. Appl, Math., 255 (2014) 544–554.

[82] J. G. Nagy, M. K. Ng, and L. Perrone *Kronecker product approximation for image restoration with reflexive boundary conditions* SIAM J. Matrix Anal. Appl., 25 (2004) 829–841.

[83] N. Nguyen, P. Milanfar and G. Golub, *Effcient Generalized Cross Validation with Applications to Parametric Image Restoration and Resolution Enhancement,* IEEE Transactions on Image Processing, 10(9) (2001) 1299-1308.

[84] A. Neuman, L. Reichel, and H. Sadok, *Algorithms for range restricted iterative methods for linear discrete ill-posed problems*, Numer. Algorithms, 59 (2012) 325–331. Code is available in Netlib at `http://www.netlbib.org/numeralgo/` in the package na33.

[85] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, Optimization (Ed. R. Fletcher), Academic Press, London, New York, (1969) 283-298.

[86] D. W. Peaceman, H. H. Rachford, *The numerical solution of parabolic and elliptic differential equations, Journal of the Society for Industrial and Applied Mathematics*, 3 (1955) 28–41.

[87] C. C. Paige and M. A. Saunders *LSQR : An algorithm for sparse linear equations and sparse least squares* ACM Trans. Math. Soft., 8 (1982) 43-71.

[88] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica 60 (1992) 259-268.

[89] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970

[90] L. Reichel and G. Rodriguez, *Old and new parameter choice rules for discrete ill-posed problems*, Numer. Algorithms, 63 (2013) 65–87.

[91] M. Rojas and D. C. Sorensen, *A trust-region approach to regularization of large-scale discrete forms of ill-posed problems*, SIAM J. Sci. Comput., 23 (2002) 1842–1860.

[92] M. Rojas and T. Steihaug, *An interior-point trust-region-based method for large-scale non-negative regularization*, Inverse Problems, 18 (2002) 1291–1307.

[93] Y. Saad, *On the Lanczos method for solving symmetric linear systems with several right-hand sides*, Math. Comp., 48 (1987) 651–662.

[94] M. Schweiger, A. Gibson and S. Arridge, *Computational Aspects of Diffuse Optical Tomography*, IEEE Computing in Science and Engineering. November/December 2003.

[95] F. Toutounian and S. Karimi, *Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides*, Appl. Math. Comput., 178 (2006) 452–460.

[96] A. Tikhonov, and V. Arsenin, *Solution of ill-posed problems*, Winston, Washington, DC, 1977.

[97] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, PA, 2002.

[98] C. R. Vogel and M. E. Oman, *Fast, robust total variation-based reconstruction of noisy blurred images*, IEEE Trans. Image Proc., 7 (1998) 813-824.

[99] C. F. Van Loan and N. P. Pitsianis, *Approximation with Kronecker products*, in Linear Algebra for Large Scale and Real Time Applications, M. S. Moonen and G. H. Golub, eds., Kluwer Publications, (1993) 293-314.

[100] S. J. Wright, M. A. T. Figueiredo, and R. D. Nowak, *Sparse Reconstruction by Separable Approximation*, IEEE Trans. Signal Processing, 57(7) (2009) 2479–2493.

[101] Y. Wang, J. Yang, W. Yin, and Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, SIAM J. Imag. Sci., 1(4) (2008) 248–272.

[102] C. L. Wu, J. Y. Zhang, and X. C. Tai, *Augmented Lagrangian method for total variation restoration with non-quadratic fidelity,* Inverse Problems and Imaging, 5 (2010) 237-261.