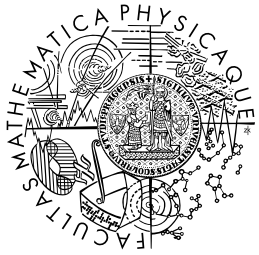


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Marie Kubínová

Numerical Methods in Discrete Inverse Problems

Department of Numerical Mathematics

Supervisor of the doctoral thesis:	RNDr. Iveta Hnětynková, Ph.D.
Study programme:	Mathematics
Specialization:	Scientific and Technical Calculations

Prague 2018

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague July 31, 2018 signature of the author

Title: Numerical Methods in Discrete Inverse Problems

Author: Marie Kubínová

Department: Department of Numerical Mathematics

Supervisor: RNDr. Iveta Hnětynková, Ph.D., Department of Numerical Mathematics

Abstract: Inverse problems represent a broad class of problems of reconstructing unknown quantities from measured data. A common characteristic of these problems is high sensitivity of the solution to perturbations in the data. The aim of numerical methods is to approximate the solution in a computationally efficient way while suppressing the influence of inaccuracies in the data, referred to as noise, that are always present. Properties of noise and its behavior in regularization methods play crucial role in the design and analysis of the methods. The thesis focuses on several aspects of solution of discrete inverse problems, in particular: on propagation of noise in iterative methods and its representation in the corresponding residuals, including the study of influence of finite-precision computation, on estimating the noise level, and on solving problems with data polluted with noise coming from various sources.

Keywords: discrete inverse problems, iterative solvers, noise estimation, mixed noise, finite-precision arithmetic

Název práce: Numerické metody pro řešení diskrétních inverzních úloh

Autor: Marie Kubínová

Katedra: Katedra numerické matematiky

Vedoucí disertační práce: RNDr. Iveta Hnětynková, Ph.D., Katedra numerické matematiky

Abstrakt: Inverzní úlohy představují širokou skupinu problémů rekonstrukce neznámých veličin z naměřených dat, přičemž společným rysem těchto problémů je vysoká citlivost řešení na změny v datech. Úkolem numerických metod je zkonstruovat výpočetně nenáročným způsobem aproximaci řešení a zároveň potlačit vliv nepřesností v datech, tzv. šumu, který je vždy přítomen. Vlastnosti šumu a jeho chování v regularizačních metodách hrají klíčovou roli při konstrukci a analýze těchto metod. Tato práce se zaměřuje na některé aspekty řešení diskrétních inverzních úloh, a to konkrétně: na propagaci šumu v iteračních metodách a jeho reprezentaci v příslušných residuích, včetně studia vlivu aritmetiky s konečnou přesností, na odhad hladiny šumu a na řešení problémů s daty zatíženými šumem z různých zdrojů.

Klíčová slova: diskrétní inverzní úlohy, iterační metody, odhadování šumu, smíšený šum, aritmetika s konečnou přesností

This thesis concludes my studies at Charles University and at this place, I would like to thank some of the people who made the work possible.

First and foremost, I am greatly indebted to my supervisor, Iveta Hnětynková, for her guidance throughout my whole studies, her persistent help and encouragement, and also for the opportunity to TA her classes at the University. I am grateful to Zdeněk Strakoš, who co-supervised my doctoral studies, for introducing me to the scientific community and for helping me to see my work in a broader perspective.

I would also like to thank people who contributed to some of the results presented in the thesis, these are: Tomáš Gergelits, Martin Plešinger, Miro Rozložník, and last but not least, Jim Nagy, who made it possible for me to come to Emory University and who supervised my work there.

During my studies, I had the privilege to meet many other great people, who made my studies a pleasant and enjoyable period of time – especially all the members of the former Coffee Club at the Institute of Computer Science, Czech Academy of Sciences, and my fellow students, with whom I happily shared the office at the Department of Numerical Mathematics.

Coming to this point would not have been possible without my husband Petr and my parents, and their love and faith in me. Thank you.

The work presented in the thesis was supported by the Czech Science Foundation project 13-06684S, by the Charles University Grant Agency project 196216, by the Specific Academic Research Project SVV-2017-260455, and through Fulbright Scholarship.

Contents

Notation

$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set of real vectors of length n
$\mathbb{R}^{m\times n}$	set of real matrices of size $m \times n$
$(\cdot, \cdot)$	Euclidean inner product
$\|\cdot\|_2$, $\|\cdot\|$	Euclidean norm
$\|\cdot\|_F$	Frobenius norm
$\text{span}\{\cdots\}$	subspace spanned by vectors
A	coefficient matrix
A^T	transpose of A
A^{-1}	inverse of A
$A^\dagger$	Moore-Penrose pseudoinverse of A
I, I_n,	identity matrix
$I_{m,n}$	$m \times n$ matrix with ones on its diagonal and zeros elsewhere
e_i	i-th column of identity matrix
$\text{diag}(b)$,	square matrix with entries of b on its diagonal and zeros elsewhere
$\text{triu}(A)$	upper triangular part of matrix A
ϵ_{mach}	machine precision

1. Introduction

Many fields of application require numerical solution of linear inverse problems. These are often represented by the system of linear algebraic equations of the form

$$b = Ax^{\text{true}} + \eta\,, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$ represents the discrete forward model and $b \in \mathbb{R}^m$ represents the measured data. The vector η denotes unknown perturbations in the data, usually referred to as *noise*, which includes rounding errors, errors of measurement etc. Given A and b, the aim is to compute a numerical approximation of the exact solution x^{true}. If the system

$$Ax \approx b \tag{1.2}$$

is incompatible and the perturbations are Gaussian, independent and identically distributed random variables with zero mean, further referred to as white noise, the problem (1.2) is typically formulated as the problem of least squares and the associated solution

$$x^{\text{LS}} = \arg\min_x \|b - Ax\| \tag{1.3}$$

is called the least-squares solution. Inverse problems of the form (1.1) arise for example in signal and image processing, geophysics, seismology, etc. For the mentioned applications, the inverse problems are typically *ill-posed*.[1] The ill-posed nature of the problem is revealed by the singular values of A, which decay gradually to zero without a noticeable gap. Thus A is ill-conditioned and the naive least-squares solution x^{LS} is due to severe amplification of noise meaningless. To compute a meaningful approximation of x^{true} some *regularization* is necessary. Regularization can take many forms, but the target of all of them is to preserve sufficient information about the exact solution, while suppressing the influence of noise.

Most commonly known regularization approaches are based on Tikhonov's regularization (Tikhonov [1963]) or on closely related spectral filtering, such as the truncated SVD, see, e.g., Hansen [1987]. Since these methods involve computation of the (partial) SVD of A, or are in other ways computationally demanding, they are usually confined to smaller problems. A common alternative to the spectral filtering methods is iterative regularization. For matrices allowing fast matrix-vector multiplication, iterative regularization is often based on Krylov subspace methods, see, e.g., Liesen and Strakoš [2013], and regularization is achieved via projection onto a Krylov subspace of small dimension. Hybrid methods combine both types of regularization. First, the original problem is projected onto a Krylov subspace, and subsequently the projected problem is further regularized using spectral filtering.

Regularization effect of the particular method is typically controlled by a regularization parameter, and its choice is crucial for the performance of the method.

[1] According to the definition of Hadamard, ill-posed problems are those for which the solution does not exist, is not unique, or is not a continuous function of the data.

Strategies for choosing regularization parameters can be divided into two groups: methods based on some a priori knowledge about noise, such as the discrepancy principle (Morozov [1966]), and methods that work without this a priori information, such as the L-curve (Hansen [1992]) or the generalized cross validation (Golub et al. [1979]). The presented thesis contributes to several aspects of numerical solution of discrete inverse problems. It comprises four chapters and the core of each chapter is represented by a peer-reviewed publication, which is for completeness accompanied by additional comments and numerical experiments included in the sections at the end.

Chapter 2 deals with iterative regularization methods based on the Golub-Kahan iterative bidiagonalization (Golub and Kahan [1965]). We investigate, for the three most common methods LSQR, LSMR, and CRAIG, the resemblance of the obtained residuals $r_k = b - Ax_k$ to the noise vector η. This is not done by constructing the residuals and comparing them to the properties of η, which are rarely known in practice, but rather by tracking the transformation of the noise vector inside the bidiagonalization process. Due to specific smoothing properties of the matrices coming from discrete inverse problems, see also Hnětynková et al. [2009], the transformation has a specific form and allows us to describe the representation of noise in the particular residuals as well as to consider the optimal stopping iteration for some of the methods. Obtained results were published in the article Hnětynková et al. [2017], which is included in the chapter. Part of the analysis in this article relies on the exact-arithmetic behavior of the bidiagonalization, therefore we show how this behavior can be simulated using finite-precision computations.

The Golub-Kahan bidiagonalization also provides an efficient way to estimate the noise level $\|\eta\|/\|Ax\|$ in the data, which is the focus of Chapter 3. The estimated noise level may then constitute an input parameter for various other methods. For some simple problems polluted with white noise, the noise estimation using the Golub-Kahan bidiagonalization has been used already in Hnětynková et al. [2009]. In the proceedings contribution Hnětynková et al. [2016], which we include in this chapter, we present an analogous technique applied to image deblurring problems corrupted by noise with various characteristics, and we assess its reliability. We also comment on the limitations of the method when applied to problems with only a few measurements.

All iterative regularization methods based on Krylov subspaces rely on the construction of well-conditioned (ideally orthonormal) bases of these subspaces. The Golub-Kahan bidiagonalization, investigated in Chapters 2–3, as well as the Lanczos tridiagonalization (Lanczos [1950]) are techniques for generating orthonormal basis using short recurrences avoiding explicit reorthogonalization against the preceeding vectors. Short recurrences represent a great reduction in the computational effort. However, in finite-precision computations, the orthogonality of the computed vectors is often quickly lost due to rounding errors. This surprisingly does not lead to a complete failure of the methods based on these iterative processes, see, e.g., Meurant and Strakoš [2006]. On the other hand, we often observe a significant delay of convergence in comparison with the exact error-free version.

Due to this delay, it may be reasonable to associate the exact-arithmetic entities with their finite-precision counterparts in later iterations. This is technically straightforward only for entities whose size decay monotonically. For other entities, such as the residuals in the Galerkin methods, for example CG (Hestenes and Stiefel [1952]) or CRAIG (Craig [1955]), the link to their exact-arithmetic counterparts is due to possible oscillations more complicated. In Chapter 4, we include proceedings contribution Gergelits et al. [2018] investigating how non-monotonic quantities from finite-precision arithmetic computations can be associated with their exact arithmetic counterparts.

In specific applications, some a priori information about the statistical distribution of noise in (1.1) may be available. The least squares formulation (1.3) of (1.1) is appropriate only for white noise, for which it represents the maximum likelihood estimate, see, e.g., Vogel [2002]. For problems with other types of noise, the objective functional has to take a different form. In image processing applications, the data often contains a combination of Poisson and additive Gaussian noise. For problems with mixed noise, we have to rely on some approximation of the likelihood functional. Staglianò et al. [2011] showed that one of the possible approximations leads to a weighted least-squares problem with solution-dependent weights. Problems for which part of the data is further influenced by severe corruptions, often referred to as outliers, in addition to noise, was to our knowledge not studied in the literature. In Chapter 5 we include article Kubínová and Nagy [in press], in which we deal with those problems and propose an objective functional combining the least squares representation with a robust loss function taking care of the outliers. We also propose two possible optimization schemes.

Chapter 6 summarizes the main ideas of the thesis and formulates open questions.

Bibliography

E. J. Craig. The N-step iteration procedures. *J. Math. Phys.*, 34:64–73, 1955.

T. Gergelits, I. Hnětynková, and M. Kubínová. Relating computed and exact entities in methods based on Lanczos tridiagonalization. In T. Kozubek, M. Čermák, P. Tichý, R. Blaheta, J. Šístek, D. Lukáš, and J. Jaroš, editors, *High Performance Computing in Science and Engineering*, pages 73–87, Cham, 2018. Springer International Publishing.

G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM: Series B, Numerical Analysis*, 2:205–224, 1965.

G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

P. C. Hansen. The truncated SVD as a method for regularization. *BIT*, 27(4): 534–553, 1987.

P. C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Rev.*, 34(4):561–580, 1992.

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.

I. Hnětynková, M. Plešinger, and Z. Strakoš. The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data. *BIT*, 49(4):669–696, 2009.

I. Hnětynková, M. Kubínová, and M. Plešinger. Notes on performance of bidiagonalization-based noise level estimator in image deblurring. In A. Handlovičová, editor, *Proceedings of the Conference Algoritmy*, pages 333–342. Slovak University of Technology in Bratislava, Publishing House of STU, 2016.

I. Hnětynková, M. Kubínová, and M. Plešinger. Noise representation in residuals of LSQR, LSMR, and CRAIG regularization. *Linear Algebra Appl.*, 533:357–379, 2017.

M. Kubínová and J. G. Nagy. Robust regression for mixed Poisson–Gaussian model. *Numerical Algorithms*, in press.

C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

J. Liesen and Z. Strakoš. *Krylov subspace methods*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2013.

G. Meurant and Z. Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

V. A. Morozov. On the solution of functional equations by the method of regularization. *Soviet mathematics – Doklady*, 7:414–417, 1966.

A. Staglianò, P. Boccacci, and M. Bertero. Analysis of an approximate model for Poisson data reconstruction and a related discrepancy principle. *Inverse Prob.*, 27(12):125003, 2011.

A. N. Tikhonov. On the solution of incorrectly put problems and the regularisation method. In *Outlines of the joint Soviet-American Symposium on Partial Differential Equations (Novosibirsk, 1963)*, pages 261–265. Acad. Sci. USSR Siberian Branch, Moscow, 1963.

C. R. Vogel. *Computational methods for inverse problems*, volume 23 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. With a foreword by H. T. Banks.

2. Noise representation in residuals of bidiagonalization-based regularization

Many stopping criteria in regularization methods for solving discrete inverse problems are based on the resemblance between the residual $b - Ax^{\text{reg}}$ and the (unknown) noise vector. For example, if noise is believed to be white noise, we may expect the residual corresponding to a good regularized solution to have the spectral properties of white noise. We explain in the article included in Section 2.1 that the opposite procedure is also possible for methods based on the Golub–Kahan iterative bidiagonalization. More precisely, we show that independently of the noise characteristic, based solely on propagation of noise through the process, we may describe the representation of noise in each of the residuals and predict which iteration will result in a residual resembling the noise vector. In Section 2.2 we comment on how the exact-arithmetic Golub–Kahan bidiagonalization can be simulated on a computer. We acknowledge the contribution of Miroslav Rozložník to Section 2.2.

2.1 Article published in Linear Algebra and its Applications

This section contains the article Hnětynková et al. [2017]. Reprinted by permission from Elsevier: *Linear Algebra and its Applications*, Hnětynková, I., Kubínová, M. & Plešinger, M.: Noise representation in residuals of LSQR, LSMR, and CRAIG regularization, copyright (2017), (doi: 10.1016/j.laa.2017.07.031).

Linear Algebra and its Applications 533 (2017) 357–379

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Noise representation in residuals of LSQR, LSMR, and CRAIG regularization

Iveta Hnětynková [a], Marie Kubínová [a,b,*], Martin Plešinger [c]

[a] *Faculty of Mathematics and Physics, Charles University, Sokolovská 83, Prague 8, Czech Republic*
[b] *Institute of Computer Science, The Czech Academy of Sciences, Pod Vodárenskou věží 2, Prague 8, Czech Republic*
[c] *Faculty of Education, Technical University of Liberec, Studentská 2, Liberec, Czech Republic*

ARTICLE INFO

Article history:
Received 13 December 2016
Accepted 27 July 2017
Available online 1 August 2017
Submitted by V. Mehrmann

MSC:
15A29
65F10
65F22

Keywords:
Ill-posed problems
Regularization
Golub–Kahan iterative bidiagonalization
LSQR
LSMR
CRAIG

ABSTRACT

Golub–Kahan iterative bidiagonalization represents the core algorithm in several regularization methods for solving large linear noise-polluted ill-posed problems. We consider a general noise setting and derive explicit relations between (noise contaminated) bidiagonalization vectors and the residuals of bidiagonalization-based regularization methods LSQR, LSMR, and CRAIG. For LSQR and LSMR residuals we prove that the coefficients of the linear combination of the computed bidiagonalization vectors reflect the amount of propagated noise in each of these vectors. For CRAIG the residual is only a multiple of a particular bidiagonalization vector. We show how its size indicates the regularization effect in each iteration by expressing the CRAIG solution as the exact solution to a modified compatible problem. Validity of the results for larger two-dimensional problems and influence of the loss of orthogonality is also discussed.

© 2017 Elsevier Inc. All rights reserved.

* Corresponding author at: Faculty of Mathematics and Physics, Charles University, Sokolovská 83, Prague 8, Czech Republic.
E-mail addresses: hnetynko@karlin.mff.cuni.cz (I. Hnětynková), kubinova@karlin.mff.cuni.cz (M. Kubínová), martin.plesinger@tul.cz (M. Plešinger).

http://dx.doi.org/10.1016/j.laa.2017.07.031
0024-3795/© 2017 Elsevier Inc. All rights reserved.

1. Introduction

In this paper we consider ill-posed linear algebraic problems of the form

$$b = Ax + \eta, \qquad A \in \mathbb{R}^{m\times n}, \qquad b \in \mathbb{R}^m, \qquad \|\eta\| \ll \|Ax\|, \tag{1}$$

where the matrix A represents a discretized smoothing operator with the singular values decaying gradually to zero without a noticeable gap. We assume that multiplication of a vector v by A or A^T results in smoothing which reduces the relative size of the high-frequency components of v. The operator A and the vector b are supposed to be known. The vector η represents errors, such as *noise*, that affect the exact data. Problems of this kind are commonly referred to as linear discrete ill-posed problems or linear inverse problems and arise in a variety of applications [1,2]. Since A is ill-conditioned, the presence of noise makes the naive solution

$$x^{\text{naive}} \equiv A^\dagger b,$$

where $A^\dagger$ denotes the Moore–Penrose pseudoinverse, meaningless. Therefore, to find an acceptable numerical approximation to x, it is necessary to use regularization methods.

Various techniques to regularize the linear inverse problem (1) have been developed. For large-scale problems, iterative regularization is a good alternative to direct regularization methods. When an iterative method is used, regularization is achieved by early termination of the process, before noise η starts to dominate the approximate solution [1]. Many iterative regularization methods such as LSQR [3–6], CRAIG [7,8], LSMR [9], and CRAIG-MR/MRNE [10,11] involve the Golub–Kahan iterative bidiagonalization [12]. Combination with an additional inner regularization (typically with a spectral filtering method) gives so-called hybrid regularization; see, for example, [4,13–15]. Various approaches for choosing the stopping criterion, playing here the role of the regularization parameter, are based on comparing the properties of the actual residual to an a priori known property of noise, such as the noise level in the Morozov's discrepancy principle [16], or the noise distribution in the cumulative residual periodogram method [17–19]. Thus understanding how noise translates to the residuals during the iterative process is of great interest.

The aim of this paper is, using the analysis of the propagation of noise in the left bidiagonalization vectors provided in [20], to study the relation between residuals of bidiagonalization-based methods and the noise vector η. Whereas in [20], white noise was assumed, here we have no particular assumptions on the distribution of noise. We only assume the amount of noise is large enough to make the noise propagation visible through the smoothing by A in construction of the bidiagonalization vectors. This is often the case in ill-posed problems, as we illustrate on one-dimensional (1D) as well as significantly noise contaminated two-dimensional (2D) benchmarks. We prove that LSQR and LSMR residuals are given by a linear combination of the bidiagonalization vectors with the

coefficients related to the amount of propagated noise in the corresponding vector. For CRAIG, the residual is only a multiple of a particular bidiagonalization vector. This allows us to prove that an approximate solution obtained in a given iteration by CRAIG applied to (1) coincides with an exact solution of the (compatible) modified problem

$$Ax = b - \tilde{\eta}, \tag{2}$$

where $\tilde{\eta}$ is a noise vector estimate constructed from the currently computed bidiagonalization vectors. These results contribute to understanding of regularization properties of the considered methods and should be considered when devising reliable stopping criteria.

Note that since LSQR is mathematically equivalent to CGLS and CGNR, CRAIG is mathematically equivalent to CGNE and CGME [21], and LSMR is mathematically equivalent to CRLS [9], then in exact arithmetic, the analysis applies also to these methods.

The paper is organized as follows. In Section 2, after a recollection of the previous results, we study the propagation of various types of noise and the influence of the loss of orthogonality on this phenomenon. Section 3 investigates the residuals of selected methods with respect to the noise contamination in the left bidiagonalization vectors and compares their properties. Section 4 discusses validity of obtained results for larger 2D problems. Section 5 concludes the paper.

Unless specified otherwise, we assume exact arithmetic and the presented experiments are performed with full double reorthogonalization in the bidiagonalization process. Throughout the paper, $\|v\|$ denotes the standard Euclidean norm of the vector v, vector e_k denotes the k-th column of the identity matrix. By $\mathcal{P}_k$, we denote the set of polynomials of degree less or equal to k. The noise level is denoted by $\delta_{\text{noise}} \equiv \|\eta\|/\|Ax\|$. By Poisson noise, we understand $b_i \sim \text{Pois}([Ax]_i)$, i.e., the right-hand side b is a Poisson random vector with the Poisson parameter Ax. The test problems were adopted from the Regularization tools [22]. For simplicity of exposition, we assume the initial approximation $x_0 \equiv 0$ throughout the paper. Generalization to $x_0 \neq 0$ is straightforward.

2. Properties of the Golub–Kahan iterative bidiagonalization

2.1. Basic relations

Given the initial vectors $w_0 \equiv 0$, $s_1 \equiv b/\beta_1$, where $\beta_1 \equiv \|b\|$, the Golub–Kahan iterative bidiagonalization [12] computes, for $k = 1, 2, \ldots$,

$$\begin{aligned} \alpha_k w_k &= A^T s_k - \beta_k w_{k-1}\,, \qquad && \|w_k\| = 1, \\ \beta_{k+1} s_{k+1} &= A w_k - \alpha_k s_k\,, && \|s_{k+1}\| = 1, \end{aligned} \tag{3}$$

until $\alpha_k = 0$ or $\beta_{k+1} = 0$, or until $k = \min(m, n)$. Vectors $s_1, \ldots, s_k$, and $w_1, \ldots, w_k$, form orthonormal bases of the Krylov subspaces $\mathcal{K}_k(AA^T, b)$ and $\mathcal{K}_k(A^TA, A^Tb)$, respectively.

In the rest of the paper, we assume that the bidiagonalization process does not terminate before the iteration $k+1$, i.e., $\alpha_l, \beta_{l+1} > 0$, $l = 1, \dots k$.

Denoting $S_k \equiv [s_1, \dots, s_k] \in \mathbb{R}^{m\times k}$, $W_k \equiv [w_1, \dots, w_k] \in \mathbb{R}^{n\times k}$ and

$$L_k \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix} \in \mathbb{R}^{k\times k}, \qquad L_{k+} \equiv \begin{bmatrix} L_k \\ e_k^T \beta_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k},$$

we can write the matrix version of the bidiagonalization as

$$A^T S_k = W_k L_k^T, \qquad A W_k = S_{k+1} L_{k+}.$$

The two corresponding Lanczos three-term recurrences

$$(AA^T) S_k = S_{k+1}(L_{k+} L_k^T), \qquad (A^T A) W_k = W_{k+1}(L_{k+1}^T L_{k+}),$$

allow us to describe the bidiagonalization vectors s_{k+1} and w_{k+1} in terms of the Lanczos polynomials as

$$s_{k+1} = \varphi_k(AA^T) b, \qquad w_{k+1} = \psi_k(A^T A) A^T b \qquad \varphi_k, \psi_k \in \mathcal{P}_k; \tag{4}$$

see [3,4,23–25]. From (4) we have that

$$s_{k+1} = \varphi_k(AA^T) b = \varphi_k(AA^T)(Ax + \eta),$$

giving

$$s_{k+1} = \left[\varphi_k(AA^T) Ax + (\varphi_k(AA^T) - \varphi_k(0))\eta\right] + \varphi_k(0)\eta. \tag{5}$$

The first component on the right-hand side of (5) can be rewritten as

$$s_{k+1}^{\mathrm{LF}} \equiv \left[\varphi_k(AA^T) Ax + (\varphi_k(AA^T) - \varphi_k(0))\eta\right] = A q_{k-1}(AA^T)\left[x + A^T \eta\right],$$

for some $q_{k-1} \in \mathcal{P}_{k-1}$. Since A has the smoothing property, then s_{k+1}^{LF} is smooth for $k \ll \min(m, n)$. Thus s_{k+1} is a sum of a low-frequency vector and the scaled noise vector η,

$$s_{k+1} = s_{k+1}^{\mathrm{LF}} + \varphi_k(0)\eta. \tag{6}$$

Note that this splitting corresponds to the low-frequency part and propagated (non-smoothed) noise part only when $\|s_{k+1}^{\mathrm{LF}}\|^2 + \|\varphi_k(0)\eta\|^2 \approx 1$. For large ks, there is a considerable cancellation between s_{k+1}^{LF} and $\varphi_k(0)\eta$, the splitting (6) still holds but it does not correspond to our intuition of an underlying smooth vector and some added scaled noise. Thus we restrict ourselves to smaller values of k.

It has been shown in [20] that whereas for s_1 (the scaled right-hand side) the noise part in (6) is small compared to the true data, for larger k, due to the smoothing property of the matrix A and the orthogonality between the vectors s_k, the noise part becomes more significant. The noise scaling factor determining the relative amplification of the non-smoothed part of noise corresponds to the constant term of the Lanczos polynomial

$$\varphi_k(0) = (-1)^k \frac{1}{\beta_{k+1}} \prod_{j=1}^{k} \frac{\alpha_j}{\beta_j} \tag{7}$$

called the *amplification factor*.[1] Its behavior for problems with white noise was studied in [20] and the analysis concludes that its size increases with k until the *noise revealing iteration* k_{rev}, where the vector s_{k+1} is dominated by the non-smoothed part of noise. Then the amplification factor decreases at least for one iteration. Note that there is no analogy for the right bidiagonalization vectors, since all vectors w_k are smoothed and the factor $\psi_k(0)$ on average grows till late iterations. A recursive relation for $\psi_k(0)$, obtained directly from (3) has the form

$$\begin{aligned} \psi_0(0) &= \frac{1}{\alpha_1 \beta_1}, \\ \psi_k(0) &= \frac{1}{\alpha_{k+1}} (\varphi_k(0) - \beta_{k+1} \psi_{k-1}(0)), \quad k = 1, 2, \ldots . \end{aligned} \tag{8}$$

2.2. Behavior of the noise amplification factor

Influence of the noise frequency characteristics. The phenomenon of noise amplification is demonstrated on the problems from [26,22]. Figs. 1b and 1c show the absolute terms of the Lanczos polynomials φ_k and ψ_k for the problem `shaw` polluted with white noise of various noise levels. For example, for the noise level 10^{-3}, the maximum of $\varphi_k(0)$ is achieved for $k = 6$, which corresponds to the observation that the vector s_7 in Fig. 1a is the most dominated by propagated noise. Obviously, the noise revealing iteration increases with decreasing noise level. The amplification factors exhibit similar behavior before the first decrease. However, the behavior of $\varphi_k(0)$ can be more complicated. In Fig. 2a for `phillips`, the sizes of the amplification factors oscillate as a consequence of the oscillations in the sizes of the spectral components of b in the left singular subspaces of A. Thus there is a partial reduction of the noise component, which influences the subsequent iterations, even before the noise revealing iteration.

Even though [20] assumed white noise, noise amplification can be observed also for other noise settings and the formulas (4)–(8) still hold. However, for high-frequency noise, there is smaller cancellation between the low-frequency component s_{k+1}^{LF} and the noise

[1] Note that in [20] a different notation was used. The Lanczos polynomial φ_k was scaled by $\|b\|$ so that $s_{k+1} = \bar{\varphi}_k(AA^T)s_1$. The vector s_{k+1} was split into $s_{k+1} = s_{k+1}^{\text{exact}} + s_{k+1}^{\text{noise}}$. In our notation, $s_{k+1}^{\text{exact}} = s_{k+1}^{\text{LF}}$, and $s_{k+1}^{\text{noise}} = \varphi_k(0)\eta$.

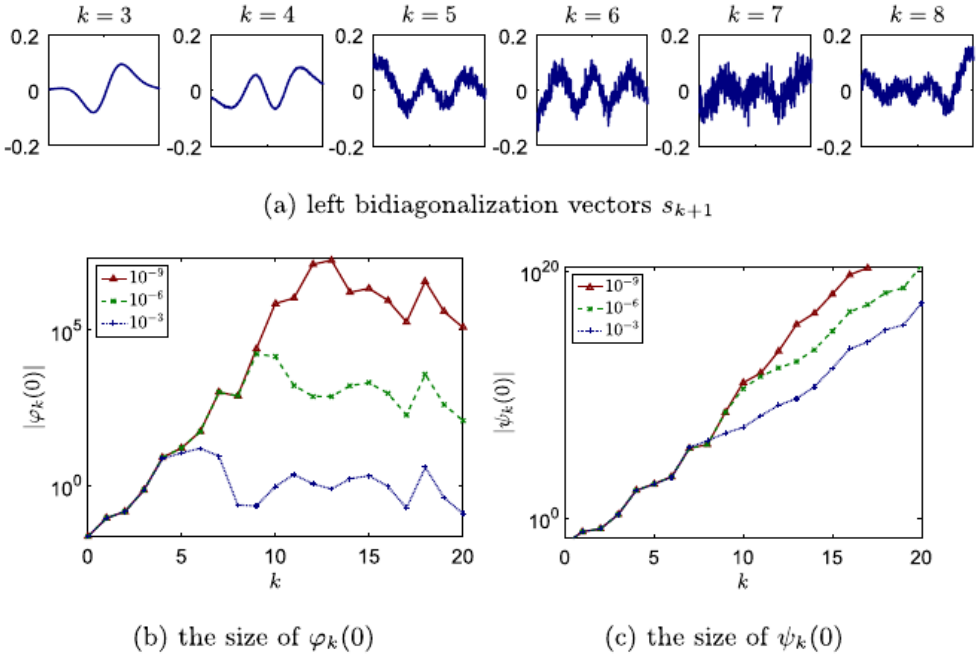


Fig. 1. The problem `shaw(400)` polluted by white noise: (a) the left bidiagonalization vectors s_{k+1} for the noise level 10^{-3}; (b) the size of the absolute term of the Lanczos polynomial φ_k for various noise levels; (c) the size of the absolute term of the Lanczos polynomial ψ_k for various noise levels.

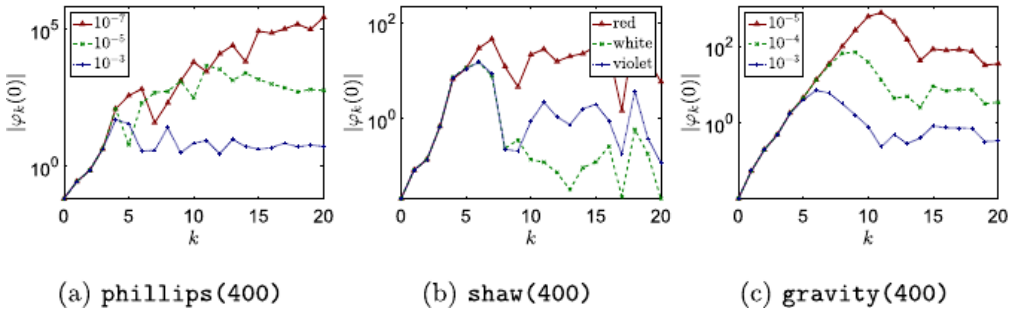


Fig. 2. Influence of the amount of noise and its frequency characteristics on the amplification factor (7): (a) the problem `phillips` with various noise levels of white noise; (b) the problem `shaw` with noise of different frequency characteristics; (c) the problem `gravity` with Poisson noise with different noise levels achieved by scaling.

part $\varphi_k(0)\eta$ in (6). Therefore, in the orthogonalization steps succeeding the noise revealing iteration k_{rev}, the noise part is projected out more significantly. For low-frequency noise, on the other side, this smoothing is less significant, which results in smaller drop of (7) after k_{rev}. This is illustrated in Fig. 2b on the problem `shaw` polluted by red (low-frequency), white, and violet (high-frequency) noise of the same noise level. For spectral characteristics of these types of noise see Fig. 3. Fig. 2c shows the amplification factor for various levels of Poisson noise.

Influence of the loss of orthogonality. First note that the splitting (6) remains valid even if φ_k are not exactly orthonormal Lanczos polynomials, since the propagated noise can

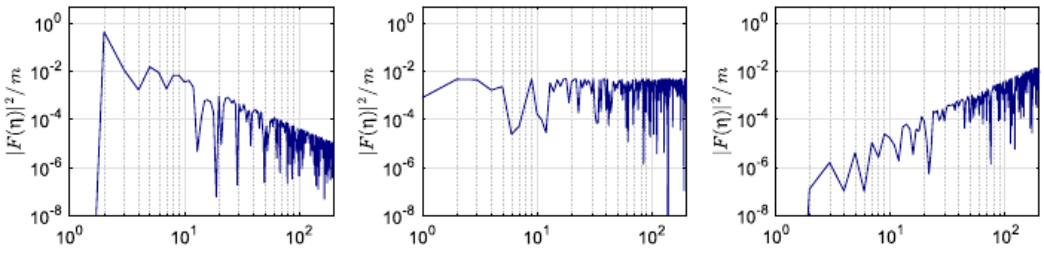


Fig. 3. Power spectral densities (or simply power spectra) for red (low-frequency dominated), white (or Gaussian), and violet (high-frequency dominated) noise η, $\|\eta\| = 1$. Power spectrum is given by squared magnitudes of Fourier coefficients $F(\eta)$ of η (see, e.g., [27, chap. 2.7]), here computed by the discrete Fourier transform. Power spectra are normalized by the length of the vector.

be still tracked using the absolute term of the corresponding (computed) polynomial. Nevertheless, it is clear that the loss of orthogonality among the left bidiagonalization vectors in finite precision arithmetic influences the behavior of the amplification factor φ_k, i.e. the propagation of noise. In the following, we denote all quantities computed without reorthogonalization by hat. Loss of orthogonality can be detected, e.g., by tracking the size of the smallest singular value $\bar{\sigma}_{\min}$ of the matrix $\hat{S}_k$ of the computed left bidiagonalization vectors. In Fig. 4 (left) for the problem `shaw` and `gravity` we see that when $\bar{\sigma}_{\min}$ drops below one detecting the loss of orthogonality among its columns, the size of the amplification factor $\hat{\varphi}_k(0)$ starts to oscillate. However, except of the delay, the larger values of $|\hat{\varphi}_k(0)|$ still match those of $|\varphi_k(0)|$. If we plot $|\hat{\varphi}_k(0)|$ against the rank of $\hat{S}_k$ instead of k, the sizes of the two amplification factors become very similar. In our experiments, the rank of $\hat{S}_k$ was computed as `rank(S(:,1:k),1e-1)` in MATLAB, i.e., singular values of $\hat{S}_k$ at least ten times smaller than they would be for orthonormal columns were considered zero. A similar shifting strategy was proposed in [28, chap. 3] for the convergence curves of the conjugate gradient method. Note that the choice of the tolerance can be problem dependent. Further study of this phenomenon is beyond the scope of this paper, but we can conclude that except of the delay the noise revealing phenomenon is in finite precision computations present.

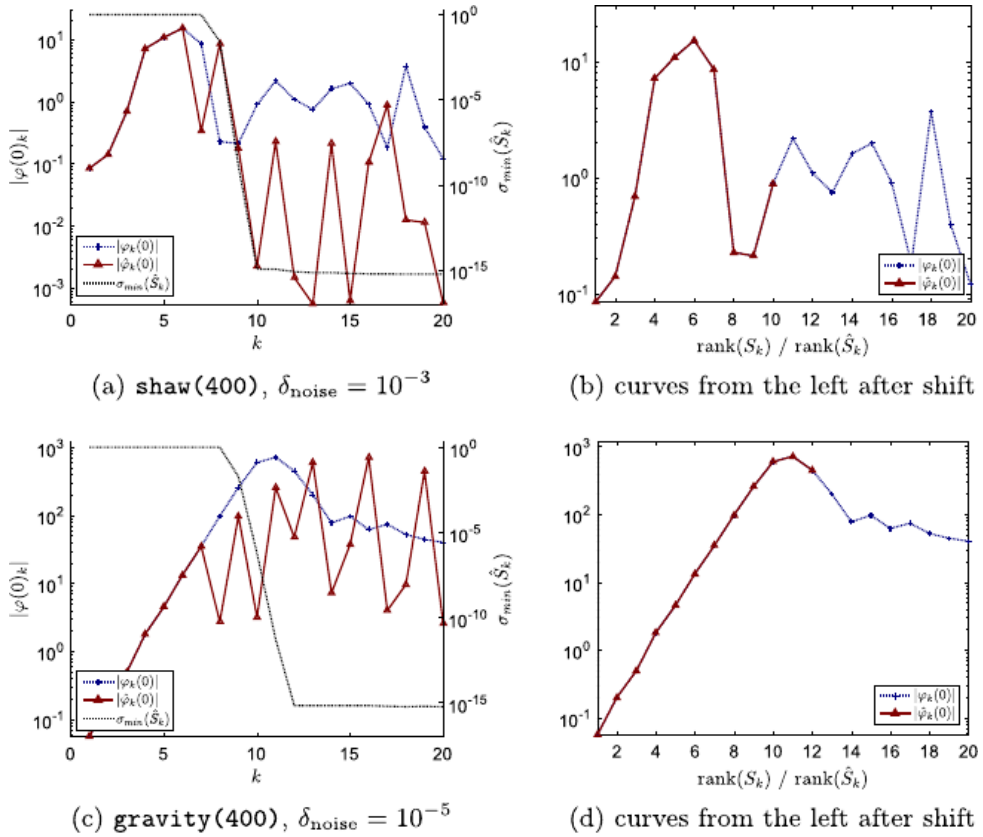
3. Noise in the residuals of iterative methods

CRAIG [7], LSQR [3], and LSMR [9] represent three methods based on the Golub–Kahan iterative bidiagonalization. At the k-th step, they search for the approximation of the solution in the subspace generated by vectors $w_1, \ldots, w_k$, i.e.,

$$x_k = W_k y_k, \qquad y_k \in \mathbb{R}^k. \tag{9}$$

The corresponding residual has the form

$$r_k \equiv b - Ax_k = b - AW_k y_k = S_{k+1}(\beta_1 e_1 - L_{k+} y_k). \tag{10}$$



(a) `shaw(400)`, $\delta_{\text{noise}} = 10^{-3}$ (b) curves from the left after shift

(c) `gravity(400)`, $\delta_{\text{noise}} = 10^{-5}$ (d) curves from the left after shift

Fig. 4. Illustration of the noise amplification for the problem `shaw` and `gravity` in finite precision computations. Left: The sizes of the amplification factor (7) computed with full double reorthogonalization ($\varphi_k(0)$) and without reorthogonalization ($\hat{\varphi}_k(0)$). Right: $|\hat{\varphi}_k(0)|$ plotted against rank($\hat{S}_k$) computed as `rank(S(:,1:k),1e-1)` in MATLAB, together with $|\varphi_k(0)|$ plotted against rank(S_k) $= k$.

CRAIG minimizes the distance of x_k from the naive solution yielding

$$L_k y_k^{\text{CRAIG}} = \beta_1 e_1. \tag{11}$$

LSQR minimizes the norm of the residual r_k yielding

$$y_k^{\text{LSQR}} = \underset{y \in \mathbb{R}^k}{\text{argmin}} \, \|\beta_1 e_1 - L_{k+} y\|. \tag{12}$$

LSMR minimizes the norm of $A^T r_k$ giving

$$y_k^{\text{LSMR}} = \underset{y \in \mathbb{R}^k}{\text{argmin}} \, \|\beta_1 \alpha_1 e_1 - L_{k+1}^T L_{k+} y\|. \tag{13}$$

These methods are mathematically equivalent to Krylov subspace methods based on the Lanczos tridiagonalization (particularly Lanczos for linear systems and MINRES) applied to particular normal equations. The relations useful in the following derivations are summarized in Table 1.

Table 1
Interpretation of bidiagonalization-based methods (CRAIG, LSQR, LSMR) as tridiagonalization-based methods (Lanczos for linear systems, MINRES) applied to the corresponding normal equations. In last two columns, the solution x of the bidiagonalization-based methods is obtained from their tridiagonalization counterparts as $x = A^T y$ and $x = A^T A z$, respectively. See also [9].

Method/equation	$(A^TA)x = A^Tb$	$(AA^T)y = b$	$(A^TA)z = A^\dagger b$
Lanczos method	LSQR(A, b)	CRAIG(A, b)	—
MINRES	LSMR(A, b)	LSQR(A, b)	CRAIG(A, b)

Since Lanczos method is a Galerkin (residual orthogonalization) method, we immediately see that

$$\begin{aligned} r_k^{\mathrm{CRAIG}} &= (-1)^k \|r_k^{\mathrm{CRAIG}}\| s_{k+1}, \\ A^T r_k^{\mathrm{LSQR}} &= (-1)^k \|A^T r_k^{\mathrm{LSQR}}\| w_{k+1}. \end{aligned} \tag{14}$$

Using the relation between the Galerkin an the residual minimization method, see [21, sec. 6.5.7], we obtain,

$$\begin{aligned} \|r_k^{\mathrm{LSQR}}\| &= \frac{1}{\sqrt{\sum_{l=0}^{k} 1/\|r_l^{\mathrm{CRAIG}}\|^2}}, \\ \|A^T r_k^{\mathrm{LSMR}}\| &= \frac{1}{\sqrt{\sum_{l=0}^{k} 1/\|A^T r_l^{\mathrm{LSQR}}\|^2}}. \end{aligned} \tag{15}$$

Note that these equations hold, up to a small perturbation, also in finite precision computations. See [29] for more details.

In the rest of this section, we investigate the residuals of each particular method. We focus on in which sense the residuals approximate the noise vector. We discuss particularly the case when noise contaminates the bidiagonalization vectors fast and thus the noise revealing iteration is well defined. More general discussion follows in Section 4.

3.1. CRAIG residuals

The following result relates approximate solution obtained by CRAIG for (1) to the solution of the problem with the same matrix and a modified right-hand side.

Proposition 1. *Consider the first k steps of the Golub–Kahan iterative bidiagonalization. Then the approximation x_k^{CRAIG} defined in (9) and (11), is an exact solution to the consistent problem*

$$Ax = b - \varphi_k(0)^{-1} s_{k+1}. \tag{16}$$

Consequently,

$$\|r_k^{\mathrm{CRAIG}}\| = |\varphi_k(0)|^{-1}. \tag{17}$$

Proof. First note that we only need to show that $r_k^{\mathrm{CRAIG}} = \varphi_k(0)^{-1} s_{k+1}$, $k = 1, 2, \ldots$. From (14) and (4) it follows that there exist $c_k \in \mathbb{R}$, such that

$$r_k^{\mathrm{CRAIG}} = c_k \cdot s_{k+1} = c_k \cdot \varphi_k(AA^T)b.$$

Let us now determine the constant c_k. From (10) and (4), we have that

$$r_k^{\mathrm{CRAIG}} = \Pi_k(AA^T)b, \quad \text{where} \quad \Pi_k \in \mathcal{P}_k \quad \text{and} \quad \Pi_k(0) = 1.$$

Combining these two equations, we obtain

$$r_k^{\mathrm{CRAIG}} = \varphi_k(0)^{-1}\varphi_k(AA^T)b. \tag{18}$$

Substituting to (18) back from (4), we immediately have (16). Since $\|s_{k+1}\| = 1$, (17) is a direct consequence of (16). □

Although the relation (16) is valid for any problem of the form (1), it has a particularly interesting interpretation for inverse problems with a smoothing operator A. Suppose we neglect the low-frequency part s_{k+1}^{LF} in (6) and estimate the unknown noise η from the left bidiagonalization vector s_{k+1} as

$$\eta \approx \tilde{\eta} \equiv \varphi_k(0)^{-1} s_{k+1}. \tag{19}$$

Subtracting $\tilde{\eta}$ from b in (1), we obtain exactly the modified problem (16). Thus Proposition 1 in fact states that in each iteration k, x_k^{CRAIG} represents the exact solution of the problem (2) with noise being approximated by a particular re-scaled left bidiagonalization vector.

The norm of the CRAIG residual r_k^{CRAIG} is inversely proportional to the amount of noise propagated to the currently computed left bidiagonalization vector. It reaches its minimum exactly in the noise revealing iteration $k = k_{\mathrm{rev}}$, which corresponds to the iteration with (19) being the best approximation of the unknown noise vector. The actual noise vector η and the difference $\eta - \tilde{\eta}$ for $\tilde{\eta}$ obtained from $s_{k_{\mathrm{rev}}+1}$ are compared in Fig. 5; see also [30]. We see that in iteration k_{rev}, the troublesome high-frequency part of noise is perfectly removed. The remaining perturbation only contains smoothed, i.e., low-frequency part of the original noise vector. The match in (17) remains valid, up to a small perturbation, also in finite precision computations, since the noise propagation is preserved, see Section 2.2.

Note that due to different frequency characteristic of η and s_{k+1}^{LF} for small k, there is a relatively small cancellation between them and

$$\|s_{k+1}^{\mathrm{LF}}\|^2 + \|\varphi_k(0)\eta\|^2 \approx 1.$$

This gives

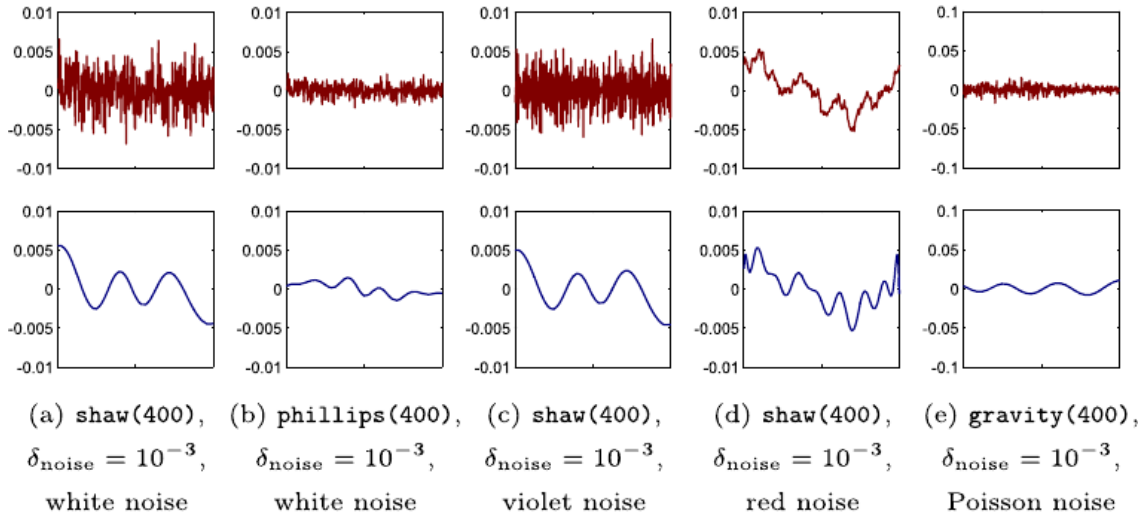


Fig. 5. Illustration of the quality of the noise vector approximation $\tilde{\eta}$ obtained by (19) for $k = k_{\text{rev}} + 1$ on various test problems and various characteristics of noise. Upper: The original noise vector η. Lower: The difference $\eta - \tilde{\eta}$.

$$\|(b - \tilde{\eta}) - Ax\| = \|\varphi_k(0)^{-1} s_{k+1}^{\text{LF}}\| \approx |\varphi_k(0)|^{-1} \sqrt{1 - \|\varphi_k(0)\eta\|^2} = \sqrt{|\varphi_k(0)|^{-2} - \|\eta\|^2}$$

supporting our expectation that the size of the remaining perturbation depends on how closely the inverse amplification factor $|\varphi_k(0)|^{-1}$ approaches $\|\eta\|$.

We may also conclude that for ill-posed problems with a smoothing operator A, the minimal error $\|x_k^{\text{CRAIG}} - x\|$ is reached approximately at the iteration with the maximal noise revealing, i.e., with the minimal residual. This is confirmed by numerical experiments in Fig. 6 comparing $\|x_k^{\text{CRAIG}} - x\|$ with $\|r_k^{\text{CRAIG}}\|$ for various test problems and noise characteristics, both with and without reorthogonalization.

3.2. LSQR residuals

Whereas for CRAIG, the residual is just a scaled left bidiagonalization vector, for LSQR it is a linear combination of all previously computed left bidiagonalization vectors. Indeed,

$$r_k^{\text{LSQR}} = b - AW_k y_k^{\text{LSQR}} = S_{k+1} \left(\beta_1 e_1 - L_{k+} y_k^{\text{LSQR}} \right), \tag{20}$$

see (10). The entries of the residual of the projected problem

$$p_k^{\text{LSQR}} \equiv \beta_1 e_1 - L_{k+} y_k^{\text{LSQR}}, \tag{21}$$

see (12), represent the coefficients of the linear combination in (20). The following proposition shows the relation between the coefficients and the amplification factor $\varphi_k(0)$.

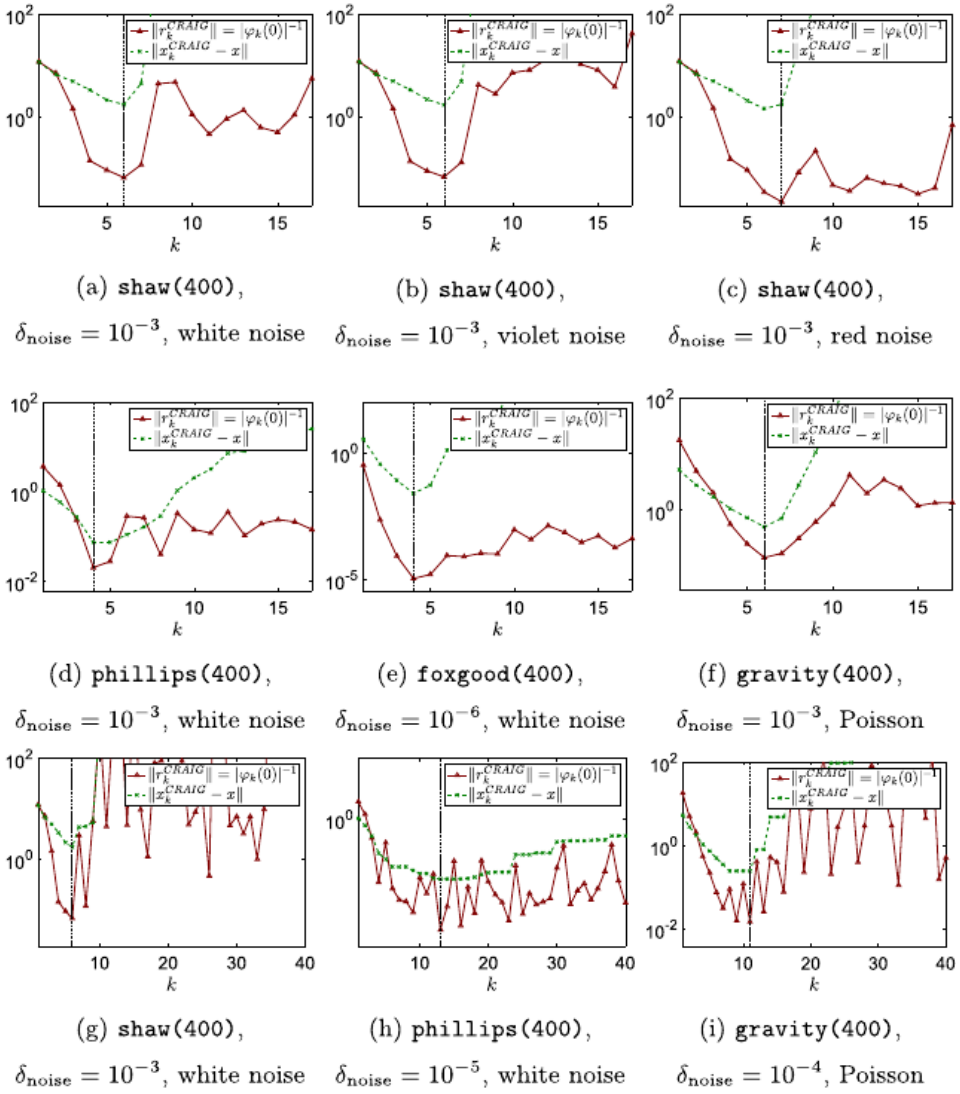


Fig. 6. Comparison of the size of the residual and the size of the error in CRAIG for various test problems with various noise characteristics. The minimal error is achieved approximately when the residual is minimized (vertical line). In Figures (g)-(i) without reorthogonalization.

Proposition 2. *Consider the first k steps of the Golub–Kahan iterative bidiagonalization. Let $r_k^{\text{LSQR}} = b - Ax_k^{\text{LSQR}}$, where x_k^{LSQR} is the approximation defined in (9) and (12). Then*

$$r_k^{\text{LSQR}} = \frac{1}{\sum_{l=0}^{k} \varphi_l(0)^2} \sum_{l=0}^{k} \varphi_l(0) s_{l+1}\,. \tag{22}$$

Consequently,

$$\|r_k^{\text{LSQR}}\| = \frac{1}{\sqrt{\sum_{l=0}^{k} \varphi_l(0)^2}}.$$

Proof. Since

$$y_k^{\mathrm{LSQR}} = \operatorname*{argmin}_y \|\beta_1 e_1 - L_{k+} y\|,$$

we get

$$L_{k+}^T p_k^{\mathrm{LSQR}} = 0.$$

It follows from the structure of the matrix L_{k+} that the entries of p_k^{LSQR} satisfy

$$\alpha_l e_l^T p_k^{\mathrm{LSQR}} + \beta_{l+1} e_{l+1}^T p_k^{\mathrm{LSQR}} = 0, \qquad \text{for} \qquad l = 1, \ldots, k.$$

Thus

$$p_k^{\mathrm{LSQR}} = c_k \begin{bmatrix} \varphi_0(0) \\ \varphi_1(0) \\ \vdots \\ \varphi_k(0) \end{bmatrix}, \tag{23}$$

where c_k is a factor that changes with k. From (15) and (18) it follows that

$$\|p_k^{\mathrm{LSQR}}\| = \|r_k^{\mathrm{LSQR}}\| = \frac{1}{\sqrt{\sum_{l=0}^{k} \varphi_l(0)^2}}. \tag{24}$$

By comparing (23) and (24), we get

$$c_k = \frac{1}{\sum_{l=0}^{k} \varphi_l(0)^2},$$

which together with (20) and (21) yields (22). □

In other words, Proposition 2 says that the coefficients of the linear combination (20) follow the behavior of the amplification factor in the sense that representation of a particular left bidiagonalization vector s_{l+1} in the residual r_k^{LSQR}, $k \geq l$, is proportional to the amount of propagated non-smoothed part of noise η in this vector.

Relation (22) also suggests that the norm-minimizing process (LSQR) and the corresponding Galerkin process (CRAIG) provide similar solutions whenever

$$\frac{\varphi_k(0)^2}{\sum_{l=0}^{k} \varphi_l(0)^2} \approx 1,$$

i.e., whenever the noise revealing in the last left bidiagonalization vector s_{k+1} is much more significant than in all previous left bidiagonalization vectors $s_1, \ldots s_k$, i.e., typically before we reach the noise revealing iteration. This is confirmed numerically in Fig. 7, comparing the semiconvergence curves of CRAIG and LSQR.

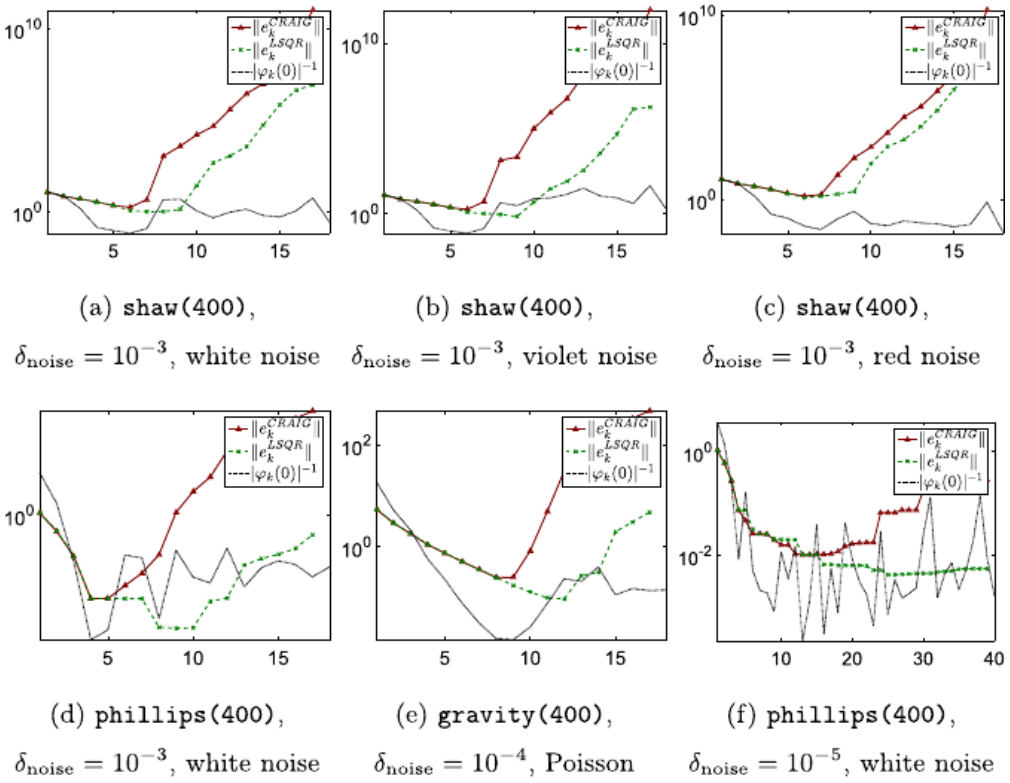


Fig. 7. The size of the error of LSQR and CRAIG in comparison with the inverse of the amplification factor for various test problems with various noise characteristics. The semiconvergence curves exhibit similar behavior until the noise revealing iteration. In Figure (f) without reorthogonalization.

3.3. LSMR residuals

Before we investigate the residual of LSMR with respect to the basis S_k, we should understand how it is related to the residual of LSQR. It follows from Table 1 that the relation between $A^T r_k^{\text{LSMR}}$ and $A^T r_k^{\text{LSQR}}$ is analogous to the relation between r_k^{CRAIG} and r_k^{LSQR}. Using Proposition 1 and 2, with φ_k substituted by ψ_k and s_k substituted by w_k, we obtain

$$A^T r_k^{\text{LSQR}} = \psi_k(0)^{-1} w_{k+1},$$

and

$$A^T r_k^{\text{LSMR}} = \frac{1}{\sum_{l=0}^{k} \psi_l(0)^2} \sum_{l=0}^{k} \psi_l(0) w_{l+1}.$$

Since

$$A^T r_k^{\text{LSMR}} = W_{k+1} L_{k+1}^T p_k^{\text{LSMR}},$$

we obtain that

$$L_{k+1}^T p_k^{\mathrm{LSMR}} = \frac{1}{\sum_{l=0}^{k} \psi_l(0)^2} \begin{bmatrix} \psi_0(0) \\ \psi_1(0) \\ \vdots \\ \psi_k(0) \end{bmatrix}. \tag{25}$$

This equality however does not provide the desired relationship between the residuals r_k^{LSMR} themselves and the left bidiagonalization vectors $s_1, \ldots, s_{k+1}$. This is given in the following proposition.

Proposition 3. *Consider the first k steps of the Golub–Kahan iterative bidiagonalization. Let $r_k^{\mathrm{LSMR}} = b - Ax_k^{\mathrm{LSMR}}$, where x_k^{LSMR} is the approximation defined in (9) and (13). Then*

$$r_k^{\mathrm{LSMR}} = \frac{1}{\sum_{l=0}^{k} \psi_l(0)^2} \sum_{l=0}^{k} \left(\varphi_l(0) \sum_{j=l}^{k} \alpha_{j+1}^{-1} \varphi_j(0)^{-1} \psi_j(0) \right) s_{l+1}.$$

Proof. From (25) it follows that

$$p_k^{\mathrm{LSMR}} = \frac{1}{\sum_{l=0}^{k} \psi_l(0)^2} L_{k+1}^{-T} \begin{bmatrix} \psi_0(0) \\ \psi_1(0) \\ \vdots \\ \psi_k(0) \end{bmatrix},$$

where L_{k+1}^{-T} is an upper triangular matrix with entries

$$e_i^T L_{k+1}^{-T} e_j = \left\{ \begin{array}{ll} \dfrac{1}{\alpha_j} & (\text{if } i = j) \\ (-1)^{i-j} \dfrac{\beta_{i+1} \cdots \beta_j}{\alpha_i \cdots \alpha_j} & (\text{if } i < j) \end{array} \right\} = \frac{\varphi_{i-1}(0)}{\alpha_j \, \varphi_{j-1}(0)}.$$

Thus

$$p_k^{\mathrm{LSMR}} = \frac{1}{\sum_{l=0}^{k} \psi_l(0)^2} \operatorname{triu} \left(\begin{bmatrix} \varphi_0(0) \\ \varphi_1(0) \\ \vdots \\ \varphi_k(0) \end{bmatrix} \begin{bmatrix} \varphi_0(0)^{-1} \\ \varphi_1(0)^{-1} \\ \vdots \\ \varphi_k(0)^{-1} \end{bmatrix}^T \right) \begin{bmatrix} \alpha_1^{-1} \psi_0(0) \\ \alpha_2^{-1} \psi_1(0) \\ \vdots \\ \alpha_{k+1}^{-1} \psi_k(0) \end{bmatrix},$$

where $\operatorname{triu}(\cdot)$ extracts the upper triangular part of the matrix. Multiplying out, we obtain

$$p_k^{\mathrm{LSMR}} = \frac{1}{\sum_{l=0}^{k} \psi_l(0)^2} \begin{bmatrix} \varphi_0(0) \sum_{l=0}^{k} \alpha_{l+1}^{-1} \varphi_l(0)^{-1} \psi_l(0) \\ \varphi_1(0) \sum_{l=1}^{k} \alpha_{l+1}^{-1} \varphi_l(0)^{-1} \psi_l(0) \\ \vdots \\ \varphi_{k-1}(0) \sum_{l=k-1}^{k} \alpha_{l+1}^{-1} \varphi_l(0)^{-1} \psi_l(0) \\ \varphi_k(0) \qquad \alpha_{k+1}^{-1} \varphi_k(0)^{-1} \psi_k(0) \end{bmatrix}. \qquad \square$$

Here the sizes of coefficients in p_k^{LSMR} need careful discussion. From (7) and (8) it follows that the absolute terms of the Lanczos polynomials $\varphi_l(0)$ and $\psi_l(0)$ have the same sign. Thus we have

$$\alpha_{l+1}^{-1} \varphi_l(0)^{-1} \psi_l(0) > 0, \quad \forall l = 0, 1, \ldots$$

and therefore the sum

$$\sum_{l=j}^{k} \alpha_{l+1}^{-1} \varphi_l(0)^{-1} \psi_l(0) \tag{26}$$

decreases when j increases. Furthermore, it was shown in [20, sec. 3.2] that for $j < k_{\mathrm{rev}}$

$$\alpha_l \approx \beta_l.$$

Thus (7) yields

$$\sum_{l=j}^{k} \alpha_{l+1}^{-1} \varphi_l(0)^{-1} \psi_l(0) \approx \sum_{l=j}^{k} \psi_l(0).$$

However, since $|\varphi_j(0)|$ on average increases rapidly with j (see Section 2.2), the sizes of the entries of p_k^{LSMR} in (3) generally increase with l before k_{rev}. After j reaches the noise revealing iteration k_{rev}, $|\varphi_j(0)|$ decreases at least for one but typically for more subsequent iterations; see Section 2.2. Multiplication by the decreasing (26) causes that the size of the entries in (3) can be expected to decrease after k_{rev}.

From the previous we conclude that the behavior of the entries of p_k^{LSMR} resembles the behavior of $\varphi_l(0)$, i.e., the size of a particular entry is proportional to the amount of propagated noise in the corresponding bidiagonalization vector, similarly as in the LSQR method. Fig. 8 compares the entries of p_k^{LSMR} with appropriately re-scaled amplification factor $\varphi_k(0)$ on the problem `shaw` with white noise. We see that the difference is negligible an therefore the residuals for LSQR and LSMR resemble. In early iterations, the resemblance of the residuals indicates resemblance of the solutions since the remaining perturbation only contains low frequencies, which are not amplified by $A^\dagger$.

Note also that since $\psi_k(0)$ grows rapidly on average, see Fig. 1c in Section 2.2, we may expect

$$\frac{\psi_k(0)^2}{\sum_{l=0}^{k} \psi_l(0)^2} \approx 1.$$

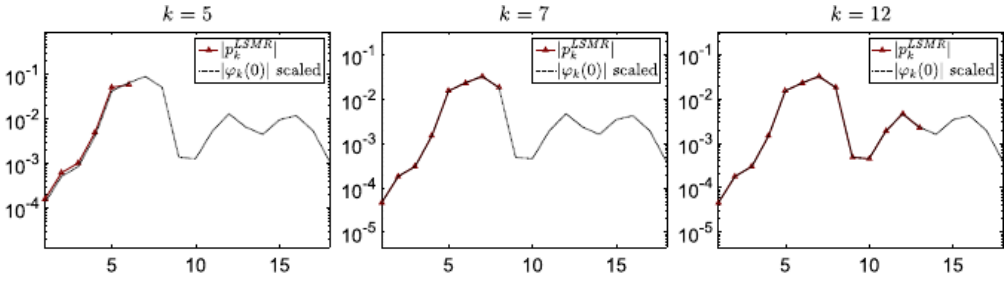


Fig. 8. The components of p_k^{LSQR} vs. the size of the amplification factor $\varphi_k(0)$ (after scaling) for several values of k for the problem `shaw` with white noise, $\delta_{\mathrm{noise}} = 10^{-3}$. The differences are negligible.

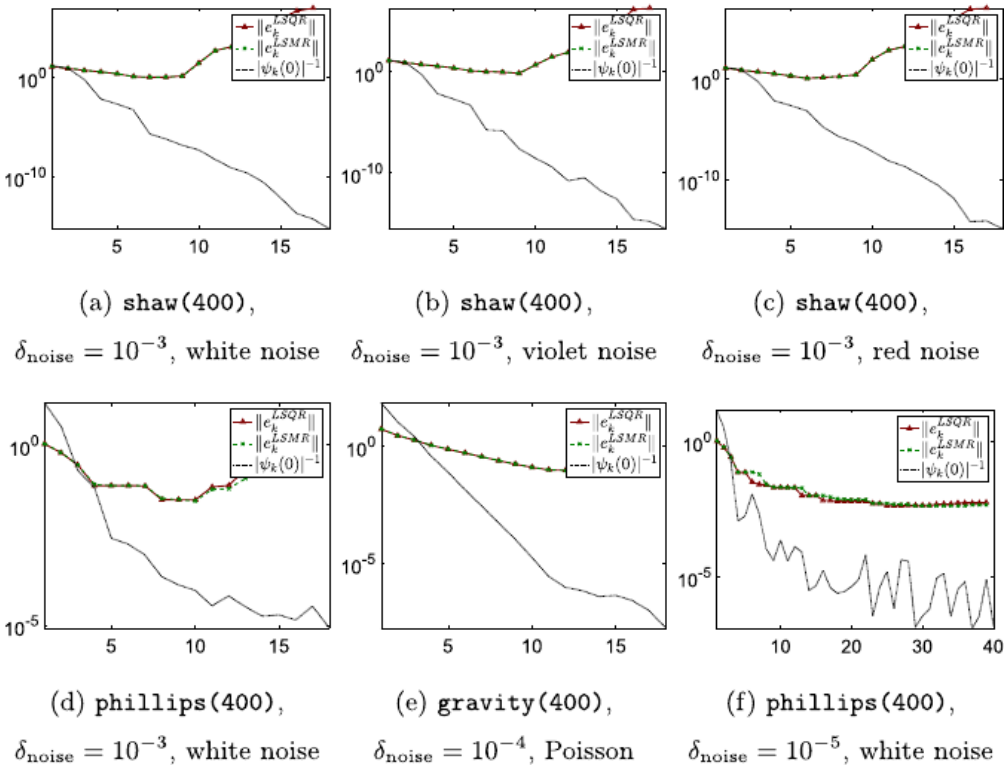
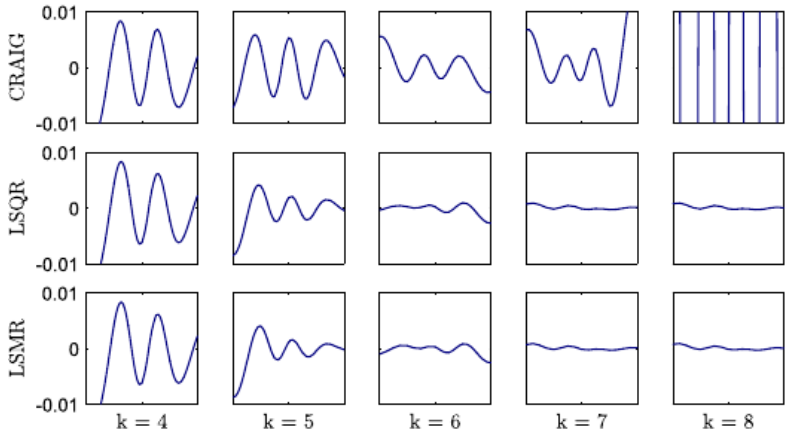


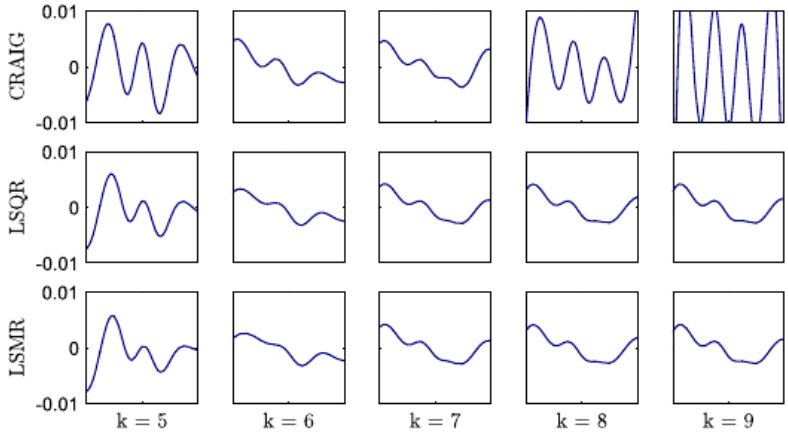
Fig. 9. The size of the error of LSMR and LSQR in comparison with the inverse of the size of $\psi_k(0)$ for various test problems with various noise characteristics. Since $|\psi_k(0)|$ often grow on average till very late iterations, the semiconvergence curves exhibit similar behavior. In Figure (f) without reorthogonalization.

Therefore $A^T r_k^{\mathrm{LSMR}}$ resembles $A^T r_k^{\mathrm{LSQR}}$ giving another explanation why LSMR and LSQR behave similarly for inverse problems with a smoothing operator A, see Fig. 9 for a comparison on several test problems.

Fig. 10 illustrates the match between the noise vector and residual of CRAIG, LSQR and LSMR method. We see that while CRAIG residual resembles noise only in the noise revealing iteration, LSQR and LSMR are less sensitive to the particular number of iterations k as the residuals are combinations of bidiagonalization vectors with appropriate coefficients. Moreover, the best match in LSQR and LSMR method overcomes the best



(a) $\eta - r_k$, white noise



(b) $\eta - r_k$, red noise

Fig. 10. Difference between the noise vector and the residual of considered iterative methods for the problem `shaw` with white and red noise noise, $\delta_{\text{noise}} = 10^{-3}$. Residuals of LSQR and LSMR have similar approximation properties with respect to the noise vector.

match in CRAIG. This is caused by the fact that the remaining low-frequency part is efficiently suppressed by the linear combination.

4. Numerical experiments for 2D problems

In this section we discuss validity of the conclusions made above for larger 2D inverse problems, where the smoothing property of A (revealing itself in the decay of singular values) is typically less significant. Consequently, noise propagation in the bidiagonalization process may be more complicated; see also [36]. However, we illustrate that essential aspects of the behavior described in previous sections are still present. Note that all experiments in this section are computed *without* reorthogonalization. We consider the following 2D benchmarks:

Medical tomography problem — a simplified 2D model of X-ray medical tomography adopted from [31], function `paralleltomo(256,0:179,362)`. The data is represented by a 256-by-256 discretization of the Shepp–Logan phantom projected in angles $\theta = 0°, 1°, \ldots, 179°$ by 362 parallel rays, resulting in a linear algebraic problem with $A \in \mathbb{R}^{65160\times 65536}$. We use Poisson-type additive noise η generated as follows (see [34, chap. 2.6] and [35]) to simulate physically realistic noise:

```
A = paralleltomo(N,theta)/N;  % forward model
t = exp(-A*x);                % transmission probabilities
c = poissrnd(t*N0);           % photon counts
eta = -log(c/N0);             % noisy measurements
```

where $N_0 = 10^5$ denotes the mean number of photons, resulting in the noise level $\delta_{\text{noise}} \approx 0.028$. We refer to this test problem as `paralleltomo`.

Seismic tomography problem — a simplified 2D model of seismic tomography adopted from [31], function `seismictomo(100,100,200)`. The data is represented by a 100-by-100 discretization of a vertical domain intersecting two tectonic plates with 100 sources located on its right boundary and 200 receivers (seismographs), resulting in a linear algebraic problem with $A \in \mathbb{R}^{20000\times 10000}$. The right-hand side is polluted with additive white noise with $\delta_{\text{noise}} = 0.01$. We refer to this test problem as `seismictomo`.

Image deblurring problem — an image deblurring problem with spatially variant blur adopted from [32,33], data `VariantGaussianBlur1`. The data is represented by a monochrome microscopic 316-by-316 image of a grain blurred by spatially variant Guassian blur (with 49 different point-spread functions), resulting in a linear algebraic problem with $A \in \mathbb{R}^{99856\times 99856}$. The right-hand side is polluted with additive white noise with $\delta_{\text{noise}} = 0.01$. We refer to this test problem as `vargaussianblur`.

Fig. 11 shows the absolute terms of the Lanczos polynomials φ_k and ψ_k. We can identify the two phases of the behavior of $\varphi_k(0)$ – average growth and average decay. However, the transition does not take place in one particular (noise revealing) iteration, but rather in a few subsequent steps, which we refer to as the *noise revealing phase* of the bidiagonalization process. The size of $\psi_k(0)$ grows on average till late iterations, however, we often observe here that the speed of this growth slows down after the noise revealing phase. In conclusion, both curves $|\varphi_k(0)|$ and $|\psi_k(0)|$ can be flatter than for 1D problem considered in previous sections. This can be further pronounced for problems with low noise levels.

Fig. 12 shows several (appropriately reshaped) left bidiagonalization vectors s_k and their cumulative periodograms for the problem `seismictomo`. Even though it is hard to make clear conclusions based on the vectors s_k themselves, we see that the periodogram for $k = 10$ is flatter than the periodograms for smaller or larger values of k, meaning that s_{10} resembles most white noise. This corresponds to Fig. 11b showing that s_{10} belongs

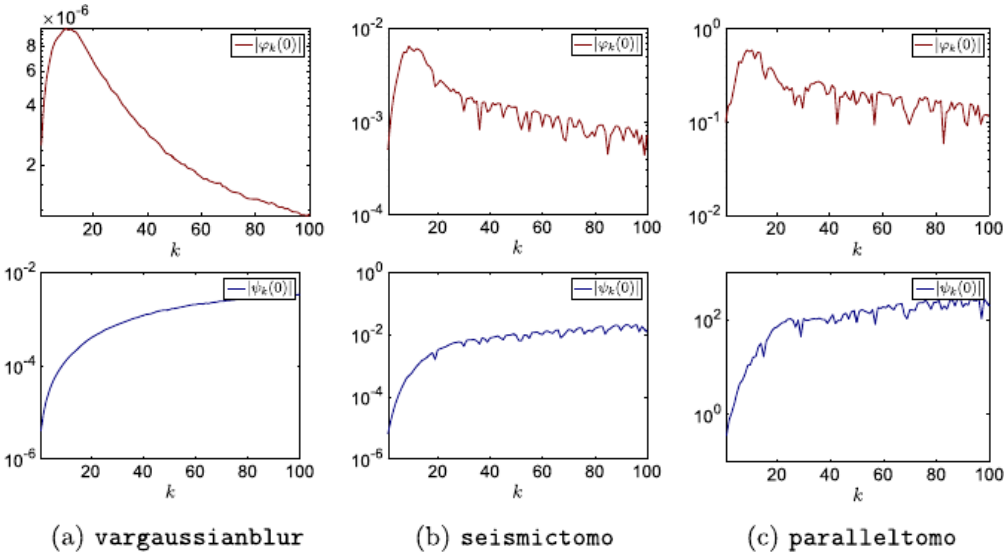


Fig. 11. The size of the absolute term of the Lanczos polynomials φ_k and ψ_k for selected 2D problems contaminated by noise as described in the text. For all problems $\delta_{\text{noise}} \approx 10^{-2}$. Computed without reorthogonalization.

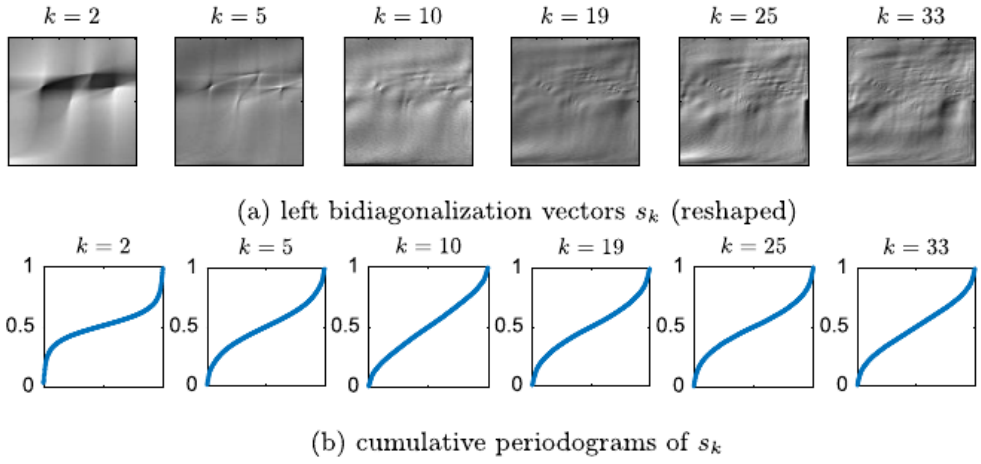


Fig. 12. Left bidiagonalization vectors s_k for the problem `seismictomo` and their cumulative periodograms. The periodogram of the vector s_{10} belonging to the noise revealing phase of the bidiagonalization process is flatter. Computed without reorthogonalization.

to the noise revealing phase of the bidiagonalization process. Note that similar flatter periodograms can be obtained for other few vectors belonging to this phase.

The absence of one particular noise revealing vector makes the direct comparison between s_k and the exact noise vector η irrelevant here. However, Propositions 1–3 remain valid and the overall behavior of the terms $|\varphi_k(0)|$ and $|\psi_k(0)|$ is as expected, allowing comparing the bidiagonalization-based methods. Fig. 13 gives comparisons of CRAIG, LSQR and LSMR for all considered 2D test problems, analogous to Fig. 6, 7, and 9. The first row of Fig. 13 shows that the CRAIG error is minimized approximately in the noise revealing phase, i.e., when the residual is minimal, see Section 3.1. The minimum is emphasized by the vertical line.

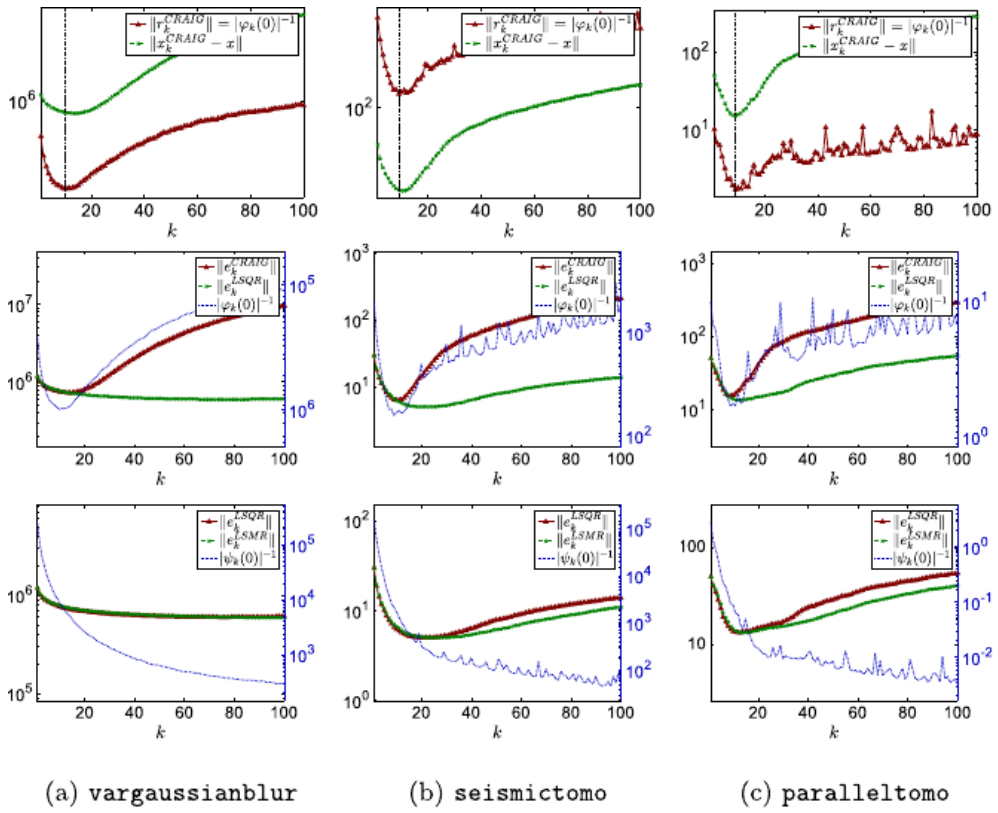


Fig. 13. First row: The size of the residual and the size of the error in CRAIG. Vertical line illustrates the minimum. Second row: The size of the error of CRAIG and LSQR, together with the rescaled inverse of the amplification factor $\varphi_k(0)$ (vertical scale on the right). Third row: The size of the error of LSQR and CRAIG, together with the rescaled inverse of the factor $\psi_k(0)$ (vertical scale on the right). Computed without reorthogonalization.

The second row of Fig. 13 compares the errors of CRAIG and LSQR. According to the derivations in Section 3.2, the curves are similar before the noise revealing phase, after which they separate with CRAIG diverging more quickly. Note that the size of the inverted amplification factor $\varphi_k(0)$ is included to illustrate the noise revealing phase and has different scaling (specified on the right).

The third row of Fig. 13 shows the errors of LSQR and LSMR with the underlying size of the inverted factor $\psi_k(0)$ (scaling specified on the right). The errors behave similarly as long as $|\psi_k(0)|^{-1}$ decays rapidly, see Section 3.3. The LSMR solution is slightly less sensitive to the particular choice of the number of bidiagonalization iterations k, which is a well know property [9].

5. Conclusion

We proved that approximating the solution of an inverse problem by the kth iterate of CRAIG is mathematically equivalent to solving consistent linear algebraic problem with the same matrix and a right-hand side, where a particular (typically high-frequency) part of noise is removed. Using the analysis of noise propagation, we showed that the size of

the CRAIG residual is given by the inverted noise amplification factor, which explains why optimal regularization properties are often obtained when the minimal residual is reached. For LSQR and LSMR, the residual is a linear combination of the left bidiagonalization vectors. The representation of these vectors in the residuals is determined by the amplification factor, in particular, left bidiagonalization vectors with larger amount of propagated noise are on average represented with a larger coefficient in both methods. These results were used in 1D problems to compare the methods in terms of matching between the residuals and the unknown noise vector. For large 2D (or 3D) problems the direct comparison of the vectors may not be possible, since noise reveals itself in a few subsequent bidiagonalization vectors (noise revealing phase of bidiagonalization) instead of in one particular iteration. However, the conclusions on the methods themselves remain generally valid. Presented results contribute to understanding of the behavior of the methods when solving noise-contaminated inverse problems.

Acknowledgements

Research supported in part by the Grant Agency of the Czech Republic under the grant 17-04150J. Work of the first and the second author supported in part by Charles University, project GAUK 196216. The authors are grateful to the anonymous referee for useful suggestions and comments that improved the presentation of the paper.

References

[1] P. Hansen, Discrete Inverse Problems, Society for Industrial and Applied Mathematics, 2010, http://dx.doi.org/10.1137/1.9780898718836.
[2] P. Hansen, Rank-Deficient and Discrete Ill-Posed Problems, Society for Industrial and Applied Mathematics, 1998, http://dx.doi.org/10.1137/1.9780898719697.
[3] C.C. Paige, M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software 8 (1) (1982) 43–71, http://dx.doi.org/10.1145/355984.355989.
[4] Å. Björck, A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations, BIT 28 (3) (1988) 659–670, http://dx.doi.org/10.1007/BF01941141.
[5] M.A. Saunders, Computing projections with LSQR, BIT 37 (1) (1997) 96–104, http://dx.doi.org/10.1007/BF02510175.
[6] T.K. Jensen, P.C. Hansen, Iterative regularization with minimum-residual methods, BIT 47 (1) (2007) 103–120, http://dx.doi.org/10.1007/s10543-006-0109-5.
[7] E.J. Craig, The N-step iteration procedures, J. Math. Phys. 34 (1955) 64–73.
[8] M.A. Saunders, Solution of sparse rectangular systems using LSQR and Craig, BIT 35 (4) (1995) 588–604, http://dx.doi.org/10.1007/BF01739829.
[9] D.C.-L. Fong, M. Saunders, LSMR: an iterative algorithm for sparse least-squares problems, SIAM J. Sci. Comput. 33 (5) (2011) 2950–2971, http://dx.doi.org/10.1137/10079687X.
[10] M. Arioli, D. Orban, Iterative Methods for Symmetric Quasi-Definite Linear Systems–Part I: Theory, 2013.
[11] K. Morikuni, K. Hayami, Inner-iteration Krylov subspace methods for least squares problems, SIAM J. Matrix Anal. Appl. 34 (1) (2013) 1–22, http://dx.doi.org/10.1137/110828472.
[12] G. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, SIAM J. Numer. Anal., Ser. B 2 (1965) 205–224.
[13] M. Hanke, On Lanczos based methods for the regularization of discrete ill-posed problems, BIT 41 (5, suppl) (2001) 1008–1018, http://dx.doi.org/10.1023/A:1021941328858, BIT 40th Anniversary Meeting.

[14] M.E. Kilmer, D.P. O'Leary, Choosing regularization parameters in iterative methods for ill-posed problems, SIAM J. Matrix Anal. Appl. 22 (4) (2001) 1204–1221, http://dx.doi.org/10.1137/S0895479899345960.
[15] J. Chung, K. Palmer, A hybrid LSMR algorithm for large-scale Tikhonov regularization, SIAM J. Sci. Comput. 37 (5) (2015) 562–580, http://dx.doi.org/10.1137/140975024.
[16] V.A. Morozov, On the solution of functional equations by the method of regularization, Sov. Math., Dokl. 7 (1966) 414–417.
[17] B.W. Rust, Parameter selection for constrained solutions to ill-posed problems, Comput. Sci. Statist. 32 (2000) 333–347.
[18] B.W. Rust, D.P. O'Leary, Residual periodograms for choosing regularization parameters for ill-posed problems, Inverse Probl. 24 (3) (2008) 034005, http://dx.doi.org/10.1088/0266-5611/24/3/034005.
[19] P.C. Hansen, M.E. Kilmer, R.H. Kjeldsen, Exploiting residual information in the parameter choice for discrete ill-posed problems, BIT 46 (1) (2006) 41–59, http://dx.doi.org/10.1007/s10543-006-0042-7.
[20] I. Hnětynková, M. Plešinger, Z. Strakoš, The regularizing effect of the Golub–Kahan iterative bidiagonalization and revealing the noise level in the data, BIT 49 (4) (2009) 669–696, http://dx.doi.org/10.1007/s10543-009-0239-7.
[21] Y. Saad, Iterative Methods For Sparse Linear Systems, 2nd edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003, http://dx.doi.org/10.1137/1.9780898718003.
[22] P.C. Hansen, Regularization tools version 4.0 for Matlab 7.3, Numer. Algorithms 46 (2) (2007) 189–194, http://dx.doi.org/10.1007/s11075-007-9136-9.
[23] G. Meurant, Z. Strakoš, The Lanczos and conjugate gradient algorithms in finite precision arithmetic, Acta Numer. 15 (2006) 471–542, http://dx.doi.org/10.1017/S096249290626001X.
[24] G. Meurant, The Lanczos and Conjugate Gradient Algorithms, Society for Industrial and Applied Mathematics, 2006, http://dx.doi.org/10.1137/1.9780898718140.
[25] G.H. Golub, G. Meurant, Matrices, Moments and Quadrature with Applications, Princeton University Press, 2009.
[26] P.C. Hansen, Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems, Numer. Algorithms 6 (1–2) (1994) 1–35, http://dx.doi.org/10.1007/BF02149761.
[27] R.G. Brown, P.Y. Hwang, Introduction to Random Signals and Applied Kalman Filtering: With Matlab Exercises and Solutions, Wiley, New York, 1997.
[28] T. Gergelits, Analysis of Krylov Subspace Methods, Master's thesis, Charles University in Prague, 2013.
[29] J. Cullum, A. Greenbaum, Relations between Galerkin and norm-minimizing iterative methods for solving linear systems, SIAM J. Matrix Anal. Appl. 17 (2) (1996) 223–247, http://dx.doi.org/10.1137/S0895479893246765.
[30] M. Michenková, Regularization Techniques Based on the Least Squares Method, Master's thesis, Charles University in Prague, 2013.
[31] P.C. Hansen, M. Saxild-Hansen, AIR-tools—a MATLAB package of algebraic iterative reconstruction methods, J. Comput. Appl. Math. 236 (8) (2012) 2167–2178, http://dx.doi.org/10.1016/j.cam.2011.09.039.
[32] S. Berisha, J.G. Nagy, Iterative Methods for Image Restoration, Academic Press Library in Signal Processing, vol. 4, 2013, pp. 193–247, Ch. 7, http://dc.doi.org/10.1016/b978-0-12-396501-1.00007-8.
[33] J.G. Nagy, K. Palmer, L. Perrone, Iterative methods for image deblurring: a Matlab object-oriented approach, Numer. Algorithms 36 (1) (2004) 73–93, http://dx.doi.org/10.1023/B:NUMA.0000027762.08431.64.
[34] T. Buzug, Computed Tomography: From Photon Statistics to Modern Cone-Beam CT, Springer-Verlag, Berlin, Heidelberg, 2008.
[35] M. S. Andersen, J. S. Jørgensen, Statistical models in X-ray computed tomography. Unpublished manuscript (February 13, 2014), DTU Compute, Technical University of Denmark.
[36] I. Hnětynková, M. Kubínová, M. Plešinger, Notes on performance of bidiagonalization-based estimator in image deblurring, in: Proceedings of Algoritmy 2016 – 20th Conference on Scientific Computing, Publishing House of Slovak University of Technology, 2016, pp. 333–342.

2.2 Simulating exact iterative bidiagonalization in finite-precision arithmetic

In some of the numerical experiments presented in Hnětynková et al. [2017] included in Section 2.1, we assumed exact arithmetic. Therefore we need to be able to simulate the exact Golub–Kahan iterative bidiagonalization on a computer. Recall that the matrix representation of the Golub–Kahan bidiagonalization has the form

$$A^T S_k = W_k L_k^T, \qquad AW_k = S_{k+1} L_{k+1,k}, \tag{2.1}$$

where the columns $s_1, \ldots, s_k$ of the matrix S_k, and the columns $w_1, \ldots, w_k$ of the matrix W_k, form orthonormal bases of the Krylov subspaces $\mathcal{K}_k(AA^T, b)$ and $\mathcal{K}_k(A^TA, A^Tb)$, respectively, see also [Hnětynková et al., 2017, sec. 2.1]. In finite-precision arithmetic, due to rounding errors, the global orthogonality among the computed vectors might be quickly lost similarly to the Lanczos method or CG. To achieve (a good level of) orthogonality among the computed vectors, *reorthogonalization* must be performed. Besides the orthogonality of the vectors, we also require that the two-term recurrences (2.1) hold within small perturbation, i.e., that the reorthogonalization terms can be absorbed into an error matrix small in norm. In the remainder of the section, to avoid excess notation, we omit higher order terms in the machine precision ϵ_{mach}. The relationship between the behavior of Krylov subspace methods in exact arithmetic and those applied to the same problem in finite-precision arithmetic is discussed also in Chapter 4 of the thesis.

The question of the size of reorthogonalization coefficients is certainly not new in the literature. For the Lanczos method, the concept of full reorthogonalization, which is reorthogonalization with respect to all previously computed Lanczos vectors, was introduced already by Lanczos [1950]. Paige [1970] claims that for the implementation there[1], under reasonable assumptions, the equation

$$B\hat{V}_k = \hat{V}_{k+1}\hat{T}_{k+1,k} + \hat{F}_k,$$

representing the matrix formulation of Lanczos process with reorthogonalization, holds with

$$\|\hat{F}_k\| \leq \mathcal{O}(n^{3/2}k^{1/2})\epsilon_{\text{mach}}\|B\|,$$

with a possible reduction in the big-O term for matrices that are very sparse or those with $\||B|\| \ll n^{1/2}\|B\|$; see also Paige [1976] or [Wilkinson, 1988, pp. 391-392]. Parlett and Scott [1979]; Parlett [1980]; Simon [1984a,b] investigated so-called semiorthogonalization, i.e., process when the loss of orthogonality is kept at the level of $\sqrt{\epsilon_{\text{mach}}}$. In bidiagonalization, there are two sets of vectors that lose orthogonality, which we want to preserve. The idea of two-sided reorthogonalization, i.e., reorthogonalization of both sets of the bidiagonalization vectors, has been appearing in literature for a long time, see, e.g., [O'Leary and Simmons,

[1]Note that Paige [1970] assumes rather nonstandard version of Lanczos with reorthogonalization, where the projections on the two preceding Lanczos vectors are computed explicitly (not using the normalization term from the previous step), and then the obtained vector is reorthogonalized against all preceding vectors. Therefore the resulting tridiagonal matrix is in this case not symmetric.

1981, p. 478–479],[Larsen, 1998, sec. 5.2],[Baglama and Reichel, 2005, p.22], and [Björck, 2014, p.289], and many others. Simon and Zha [2000] introduced the concept of one-sided reorthogonalization, i.e., reorthogonalization of only one set of vectors. Barlow [2013] investigated the backward error bound of this one-sided and one-sided selective reorthogonalization.

In the cited literature, the small size of the reorthogonalization coefficients is either given without proof or relies heavily on some previous work, with possibly different implementation of the algorithm, which makes the proofs somewhat difficult to follow. In this section we present the reorthogonalization strategy that was used in Hnětynková et al. [2017] to simulate exact-arithmetic iterative bidiagonalization including the discussion of the level of the loss of orthogonality and the validity of the obtained two-term recurrences. Note that for this purpose, we are not concerned with the computational efficiency of the proposed method.

2.2.1 Connection to Lanczos tridiagonalization

Instead of investigating the bidiagonalization itself, it is more convenient to look at the related Lanczos process. In exact arithmetic, we may relate the k-th step of the Golub–Kahan iterative bidiagonalization to:

- the k-th and $(k-1)$-st step of the two independent Lanczos processes as

$$(AA^T)S_k = S_{k+1}(L_{k+1,k}L_k^T) \quad \text{and} \quad (A^TA)W_{k-1} = W_k(L_k^TL_{k,k-1}); \tag{2.2}$$

- the $2k$-th step of the Lanczos process with an extended matrix as

$$\overbrace{\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}}^{B} \overbrace{\left[\begin{bmatrix} s_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ w_1 \end{bmatrix}, \begin{bmatrix} s_2 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ w_k \end{bmatrix}\right]}^{V_{2k}} =$$

$$= \underbrace{\left[\begin{bmatrix} s_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ w_1 \end{bmatrix}, \begin{bmatrix} s_2 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} s_{k+1} \\ 0 \end{bmatrix}\right]}_{V_{2k+1}} \underbrace{\begin{bmatrix} 0 & \alpha_1 & & & & \\ \alpha_1 & \ddots & \beta_2 & & & \\ & \beta_2 & \ddots & \alpha_2 & & \\ & & \alpha_2 & \ddots & \ddots & \\ & & & \ddots & \ddots & \alpha_k \\ & & & & \alpha_k & 0 \\ & & & & & \beta_{k+1} \end{bmatrix}}_{T_{2k+1,2k}}; \tag{2.3}$$

see also [Larsen, 1998, sec. 3.3.2]. Since in the standard Golub-Kahan bidiagonalization, the two sets of bidiagonalization vectors are computed simultaneously, representation (2.2) is of little use for the round-off error analysis in

finite-precision computation. Therefore we further use the representation (2.3). To simplify the notation, we define

$$N \equiv \max(m, n).$$

2.2.2 Reorthogonalization

Reorthogonalization procedures, to prevent severe loss of orthogonality, further orthogonalize the new vector computed by the short recurrence explicitly against (some of) the preceding vectors. Instead of the two-term recurrences, one obtains (overloading the notation from the exact arithmetic)

$$\begin{aligned}
\tilde{w}_k &= A^T s_k - \beta_k w_{k-1} - f'_{w_k}, \\
\alpha_k w_k &= \tilde{w}_k - \sum_{j=1}^{k-1} \xi_{w_{k,j}} w_j - f''_{w_k}, \\
\tilde{s}_{k+1} &= A w_k - \alpha_k s_k - f'_{s_{k+1}}, \\
\beta_{k+1} s_{k+1} &= \tilde{s}_{k+1} - \sum_{j=1}^{k} \xi_{s_{k+1,j}} s_j - f''_{s_{k+1}},
\end{aligned} \tag{2.4}$$

where ξs are the reorthogonalization coefficients that depend on the particular reorthogonalization technique, and fs represent local rounding errors. In matrix form, equations (2.4) become

$$\overbrace{\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}}^{B} \overbrace{\left[\begin{bmatrix} s_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ w_1 \end{bmatrix}, \begin{bmatrix} s_2 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ w_k \end{bmatrix}\right]}^{V_{2k}} =$$

$$= \underbrace{\left[\begin{bmatrix} s_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ w_1 \end{bmatrix}, \begin{bmatrix} s_2 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} s_{k+1} \\ 0 \end{bmatrix}\right]}_{V_{2k+1}} \underbrace{\begin{bmatrix} 0 & \alpha_1 + \xi_{s_{2,1}} & 0 & \xi_{s_{3,1}} & 0 & \cdots \\ \alpha_1 & 0 & \beta_2 + \xi_{w_{2,1}} & 0 & \ddots & \ddots \\ & \beta_2 & 0 & \alpha_2 + \xi_{s_{3,2}} & \ddots & \ddots \\ & & \alpha_2 & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \alpha_k + \xi_{s_{k+1,k}} \\ & & & & \alpha_k & 0 \\ & & & & & \beta_{k+1} \end{bmatrix}}_{H_{2k+1,2k} = T_{2k+1,2k} + R_{2k+1,2k}} +$$

$$+ \underbrace{\left[\begin{bmatrix} 0 \\ f_{w_1} \end{bmatrix}, \begin{bmatrix} f_{s_2} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ f_{w_2} \end{bmatrix}, \cdots, \begin{bmatrix} f_{s_{k+1}} \\ 0 \end{bmatrix}\right]}_{F_{2k}}, \tag{2.5}$$

where

$$f_{w_j} = f'_{w_j} + f''_{w_j}, \quad f_{s_j} = f'_{s_j} + f''_{s_j}.$$

Matrix $H_{2k,2k+1}$ is upper Hessenberg and is a sum of the tridiagonal $T_{2k,2k+1}$ and the strictly upper triangular $R_{2k,2k+1}$ containing the reorthogonalization coefficients, i.e.,

$$R_{2k+1,2k} = \sum_{i=1}^{k} \sum_{j=1}^{i} (\xi_{w_{i,j}} e_{2j} e_{2i+1}^T + \xi_{s_{i+1,j}} e_{2j-1} e_{2i}^T).$$

We expect the error matrix F_{2k} to be bounded as

$$\|F_{2k}\| \leq \mathcal{O}(k^{3/2}N + k^{1/2}N^{3/2})\epsilon_{\text{mach}}\|A\|; \tag{2.6}$$

see [Rozložník, 1997, chap. 3]. Note that the Golub–Kahan iterative bidiagonalization with any reorthogonalization technique can be represented by equation (2.5). Moreover it is both theoretically and computationally equivalent to the Lanczos tridiagonalization with the matrix B and the starting vector v_1, under the assumption that the same reorthogonalization scheme is used.[2]

Assume that the reorthogonalization coefficients ξ are chosen such that the loss of orthogonality is kept at the level of machine precision. This can be achieved for example by two-sided full reorthogonalization using iterated classical Gram-Schmidt (ICGS2), where we get

$$\|V_{l+1}^T V_{l+1} - I\| \leq \mathcal{O}(l^{3/2}N)\epsilon_{\text{mach}}; \tag{2.7}$$

see [Giraud et al., 2005, Theorem 2] for more details.[3] The pseudocode for iterative bidiagonalization using reorthogonalization by ICGS2 in Algorithm 1.

Algorithm 1 Bidiagonalization with two-sided full reorthogonalization by ICGS2

$s_1 \equiv b/\beta_1$, $\beta_1 \equiv \|b\|$, $w_0 \equiv 0$
for $k = 1, 2, \ldots$ **do**
  $\tilde{w}_k = A^T s_k - \beta_k w_{k-1}$
  for $i = 1, 2$ **do** ▷ full double reorthogonalization
    $\xi_{w_k}^{(i)} = W_{k-1}^T \tilde{w}_k$
    $\tilde{w}_k = \tilde{w}_k - W_{k-1}\xi_{w_k}^{(i)}$
  end for
  $\xi_{w_k} = \xi_{w_k}^{(1)} + \xi_{w_k}^{(2)}$ ▷ store the reorthogonalization coefficients
  $\alpha_k = \|\tilde{w}_k\|$
  $w_k = \tilde{w}_k/\alpha_k$
  $\tilde{s}_{k+1} = Aw_k - \alpha_k s_k$
  for $i = 1, 2$ **do** ▷ full double reorthogonalization
    $\xi_{s_{k+1}}^{(i)} = S_k^T \tilde{s}_{k+1}$
    $\tilde{s}_{k+1} = \tilde{s}_{k+1} - S_k \xi_{s_{k+1}}^{(i)}$
  end for
  $\xi_{s_{k+1}} = \xi_{s_{k+1}}^{(1)} + \xi_{s_{k+1}}^{(2)}$ ▷ store the reorthogonalization coefficients
  $\beta_{k+1} = \|\tilde{s}_k\|$
  $s_{k+1} = \tilde{s}_{k+1}/\beta_{k+1}$
end for

From (2.7) we immediately have that

$$\|V_{l+1}^T V_{l+1}\| \leq 1 + \mathcal{O}(l^{3/2}N)\epsilon_{\text{mach}} \quad \text{and} \quad \|(V_{l+1}^T V_{l+1})^{-1}\| \leq 1 + \mathcal{O}(l^{3/2}N)\epsilon_{\text{mach}},$$

[2] This is true under the most reasonable assumption that multiplication by zero is performed exactly.

[3] This holds if $\kappa([[b^T, 0]^T, BV_l - V_l \text{triu}(T_l)]) \cdot \epsilon_{\text{mach}} \ll 1$, i.e., before $\|w_k\|$ or $\|s_{k+1}\|$ becomes negligible.

and the norm of the Hessenberg matrix becomes bounded by

$$\begin{aligned}\|H_{l+1,l}\| &\leq \|(V_{l+1}^T V_{l+1})^{-1}\|\|V_{l+1}^T\| \left(\|B\|\|V_l\| + \|F_l\|\right) \\ &\leq (1 + \mathcal{O}(l^{3/2}N + l^{1/2}N^{3/2}))\epsilon_{\text{mach}}\|A\|. \end{aligned} \tag{2.8}$$

We now show that the reorthogonalization terms stored in $R_{l+1,l}$ are negligible and the term $V_{l+1}R_{l+1,l} = V_l R_l$ can be included in the error matrix F_l. The intuition is that since B is symmetric and V_{l+1} is almost orthogonal, H_l also needs to be almost symmetric. We do this by combining a couple of matrix inequalities. By multiplying (2.5) from the left by V_l^T and using $H_{l+1,l} = T_{l+1,l} + R_{l+1,l}$, we obtain

$$\begin{aligned} V_l^T B V_l &= V_l^T V_{l+1} H_{l+1,l} + V_l^T F_l \\ &= T_l + R_l + (V_l^T V_{l+1} - I_{l,l+1}) H_{l+1,l} + V_l^T F_l. \end{aligned} \tag{2.9}$$

Note that both B and T_l are symmetric matrices, therefore after subtracting from (2.9) its transpose, we obtain

$$\begin{aligned} R_l - R_l^T = H_{l+1,l}^T (V_l^T V_{l+1} - I_{l,l+1})^T - (V_l^T V_{l+1} - I_{l,l+1}) H_{l+1,l} + \\ + F_l^T V_l - V_l^T F_l. \end{aligned}$$

Taking norm on the both sides and using (2.6), (2.7), and (2.8), the size of the left-hand side becomes bounded as

$$\begin{aligned}\|R_l - R_l^T\| &\leq 2\left(\|V_l^T V_{l+1} - I_{l,l+1}\|\|H_{l+1,l}\| + \|V_l\|\|F_l\|\right) \\ &\leq \mathcal{O}(l^{3/2}N + l^{1/2}N^{3/2})\epsilon_{\text{mach}}\|A\|. \end{aligned}$$

We now need to estimate $\|R_l\|$ using $\|R_l - R_l^T\|$. This is possible using the norm of the Hadamard triangular truncation operator $\mathcal{T}_H$, see Angelos et al. [1992], as

$$\|R_l\| = \|\mathcal{T}_H(R_l - R_l^T)\| \leq \|\mathcal{T}_H\|\|(R_l - R_l^T)\| \leq \frac{\log N + \pi + 1}{\pi}\|R_l - R_l^T\|.$$

Since $\log N$ is negligible compared to any power of N, we will not include it into the big-O term and have

$$\|R_l\| \leq \mathcal{O}(l^{3/2}N + l^{1/2}N^{3/2})\epsilon_{\text{mach}}\|A\|,$$

and finally

$$\|BV_l - V_{l+1}T_{l+1,l}\| \leq \|V_l\|\|R_l\| + \|F_l\| \leq \mathcal{O}(l^{3/2}N + l^{1/2}N^{3/2})\epsilon_{\text{mach}}\|A\|.$$

Besides being small in norm, the matrix $V_l R_l$ has the same nonzero structure as the matrix F_l allowing us to rearrange (2.5) to the form of bidiagonalization process for which we have

$$\begin{aligned}\|AW_k - S_{k+1}L_{k+1,k}\| &\leq \mathcal{O}(k^{3/2}N + k^{1/2}N^{3/2})\epsilon_{\text{mach}}\|A\|\,, \\ \|A^T S_k - W_k L_k^T\| &\leq \mathcal{O}(k^{3/2}N + k^{1/2}N^{3/2})\epsilon_{\text{mach}}\|A\|\,. \end{aligned}$$

This shows that after the two-sided full double reorthogonalization, the equations describing matrix formulation of the bidiagonalization process (2.1) indeed hold within a perturbation of the level of $\epsilon_{\text{mach}}\|A\|$. Note also that the bound (2.6) is

derived for a general case, where the orthogonalization coefficients ξ are proportional to the size of the vector that is orthogonalized. Since we showed that the reorthogonalization coefficient ξ must be of order of ϵ_{mach} we may expect the size of the error matrix to be reduced to

$$\|F_{2k}\| \leq \mathcal{O}(k^{1/2}N^{3/2})\epsilon_{\text{mach}}\|A\|.$$

In the subsequent analysis, we however still remain limited by the level of the loss of orthogonality (2.7).

Bibliography

J. R. Angelos, C. C. Cowen, and S. K. Narayan. Triangular truncation and finding the norm of a Hadamard multiplier. *Linear Algebra Appl.*, 170:117–135, 1992.

J. Baglama and L. Reichel. Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM J. Sci. Comput.*, 27(1):19–42, 2005.

J. L. Barlow. Reorthogonalization for the Golub-Kahan-Lanczos bidiagonal reduction. *Numerische Mathematik*, 124(2):237–278, 2013.

Å. Björck. Stability of two direct methods for bidiagonalization and partial least squares. *SIAM J. Matrix Anal. Appl.*, 35(1):279–291, 2014.

L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof. Rounding error analysis of the classical Gram-Schmidt orthogonalization process. *Numerische Mathematik*, 101(1):87–100, 2005.

I. Hnětynková, M. Kubínová, and M. Plešinger. Noise representation in residuals of LSQR, LSMR, and CRAIG regularization. *Linear Algebra Appl.*, 533:357–379, 2017.

C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

R. M. Larsen. Lanczos bidiagonalization with partial reorthogonalization. *DAIMI Report Series*, 27(537), 1998.

D. P. O'Leary and J. A. Simmons. A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. *SIAM Journal on Scientific and Statistical Computing*, 2(4):474–489, 1981.

C. C. Paige. Practical use of the symmetric Lanczos process with reorthogonalization. *Nordisk tidskrift for informationsbehandling (BIT)*, 10:183–195, 1970.

C. C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *IMA Journal of Applied Mathematics*, 18(3):341–349, 1976.

B. N. Parlett. *The symmetric eigenvalue problem*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980. Prentice-Hall Series in Computational Mathematics.

B. N. Parlett and D. S. Scott. The Lanczos algorithm with selective orthogonalization. *Math. Comput.*, 33(145):217–238, 1979.

M. Rozložník. *Numerical stability of the GMRES method*. PhD thesis, Czech Technical University, 1997.

H. D. Simon. Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. *Linear Algebra Appl.*, 61:101–131, 1984a.

H. D. Simon. The Lanczos algorithm with partial reorthogonalization. *Math. Comp.*, 42(165):115–142, 1984b.

H. D. Simon and H. Zha. Low-rank matrix approximation using the Lanczos bidiagonalization process with applications. *SIAM J. Sci. Comput.*, 21(6): 2257–2274, 2000.

J. H. Wilkinson. *The algebraic eigenvalue problem*. Monographs on Numerical Analysis. The Clarendon Press, Oxford University Press, New York, 1988. Oxford Science Publications.

3. Estimating noise level through Golub-Kahan bidiagonalization

Since many numerical algorithms for solving discrete inverse problems rely on some a priori knowledge about the size of noise present in the data, estimation of the noise level remains a very active field of research. As investigated in Hnětynková et al. [2009], the Golub–Kahan bidiagonalization may provide a very cheap way of estimating the noise level for some discrete inverse problems. This observation was supported by experiments on small one-dimensional severely ill-posed problems with a square matrix and data polluted with white noise. In this chapter, we investigate the performance of the estimator on large 2D image deblurring problems with data polluted with noise of various characteristics. In Section 3.2, we further study changes in the performance of the estimator when moving from square to rectangular matrices.

3.1 Contribution in Proceedings of Algoritmy conference

This section contains the contribution Hnětynková et al. [2016] published in peer-reviewed Proceedings of the conference Algoritmy.

Proceedings of ALGORITMY 2016
pp. 333–342

NOTES ON PERFORMANCE OF BIDIAGONALIZATION-BASED NOISE LEVEL ESTIMATOR IN IMAGE DEBLURRING *

IVETA HNĚTYNKOVÁ†, MARIE KUBÍNOVÁ‡, AND MARTIN PLEŠINGER§

Abstract. Image deblurring represents one of important areas of image processing. When information about the amount of noise in the given blurred image is available, it can significantly improve the performance of image deblurring algorithms. The paper [11] introduced an iterative method for estimating the noise level in linear algebraic ill-posed problems contaminated by white noise. Here we study applicability of this approach to image deblurring problems with various types of blurring operators. White as well as data-correlated noise of various sizes is considered.

Key words. image deblurring, linear ill-posed problem, noise, noise level estimate, Golub–Kahan iterative bidiagonalization

AMS subject classifications. 15A29, 65F10, 65F22

1. Introduction.

When recording digital images, some form of blurring often occurs, e.g., when camera lens is out of focus, light conditions are not perfect, the object is moving etc. In such a case the information from a particular image pixel is spread to surrounding pixels resulting in a lower-quality image. As an additional problem, the recorded image can contain unknown errors in form of variations of pixel density usually referred to as noise with different properties based on its origin. Image deblurring methods aim to reconstruct the true sharp image while suppressing the influence of noise, by using a mathematical model of the blurring process; see, e.g., [8, Chapters 1 and 3].

Let B, $X \in \mathbb{R}^{l\times m}$ represent the blurred noisy image and its unknown sharp counterpart, respectively. In many applications, the blurring process is linear or can be well approximated by a linear model, which is an assumption we will follow. In that case we can model the blurring process as

$$Ax \approx b\,, \qquad A \in \mathbb{R}^{n\times n}\,, \qquad b \in \mathbb{R}^{n}\,, \tag{1.1}$$

where A is a linearized (e.g., discretized) blurring operator, b and x are vectorized forms of B and X (obtained by stacking the columns of the matrix into a single vector), respectively, and $n = lm$. The right-hand side of the linear algebraic problem above can be formally written as

$$b = b^{\text{exact}} + b^{\text{noise}},$$

where b^{exact} is the unknown smooth noise-free right-hand side and b^{noise} represents unknown noise. We refer to the quantity

$$\delta_{\text{noise}} \equiv \frac{\|b^{\text{noise}}\|}{\|b^{\text{exact}}\|} \tag{1.2}$$

*This work has been supported by the GAČR grant No. P201/13–06684S.

†Charles University in Prague, Faculty of Mathematics and Physics, Prague, and Institute of Computer Science, AS CR, Prague, Czech Republic (hnetynkova@cs.cas.cz).

‡Charles University in Prague, Faculty of Mathematics and Physics, Prague, and Institute of Computer Science, AS CR, Prague, Czech Republic (kubinova@cs.cas.cz).

§Department of Mathematics, Technical University of Liberec, Liberec, and Institute of Computer Science, AS CR, Prague, Czech Republic (martin.plesinger@tul.cz).

as the *noise level*. Since noise is supposed to be small compared to noise-free data, $\delta_{\text{noise}} \ll 1$ is a realistic assumption.

Properties of this model have been widely analyzed; see [6] for a summary, [14] and [8] for applications in image processing, and also [11], [3], [10] for the behavior in the context of Krylov subspace methods. In particular, it is known that the singular values of A usually decay gradually to zero without a noticeable gap and the singular vectors of A represent increasing frequencies. The model satisfies the discrete version of the so-called *Picard condition* meaning that on average the sizes of projections of b^{exact} onto the left singular subspaces of A decrease faster than the singular values. On the other hand, b^{noise} typically does not satisfy such a condition. Consequently, linear image deblurring models (1.1) represent a typical example of an *ill-posed problem*; see [8, Chapter 5].

It is well known that information about the amount of noise can significantly improve performance of image deblurring methods; see for example [13], [2], [1], [7], and also [8, Chapter 6]. Such information is however rarely available. The paper [11] introduced an inexpensive method for estimating the unknown white noise level in linear ill-posed algebraic problems with a smoothing operator A. The estimate is obtained by the Golub–Kahan iterative bidiagonalization [4], a short recurrence based Krylov subspace method. It relies on the assumptions that the model (1.1) satisfies the discrete Picard condition, A has the smoothing property, the left singular vectors of A represent increasing frequencies, and b^{exact} is smooth. Because the method needs only evaluation of matrix-vector products, it can take advantage of a specific structure of A often present in image deblurring problems; see [8, Chapter 4].

The paper is organized as follows. Section 2 summarizes the main ideas of the noise level estimation presented in [11]. Section 3 studies its applicability to image deblurring problems with various types and amount of noise. Spatially invariant as well as spatially variant blur is considered. Section 4 concludes the paper.

2. Iterative noise level estimate. The Golub–Kahan iterative bidiagonalization starting with the vectors $w_0 \equiv 0$, $s_1 \equiv b/\beta_1$, where $\beta_1 \equiv \|b\| \neq 0$, computes for $j - 1, 2, \ldots$

$$\begin{aligned}
&\widetilde{w}_j = A^T s_j - \beta_j w_{j-1} && \text{(orthogonalization step)}\\
&\alpha_j = \|\widetilde{w}_j\|\,, \qquad w_j = \frac{1}{\alpha_j}\,\widetilde{w}_j && \text{(normalization step)}\\
&\widetilde{s}_{j+1} = A w_j - \alpha_j s_j && \text{(orthogonalization step)}\\
&\beta_{j+1} = \|\widetilde{s}_{j+1}\|\,, \quad s_{j+1} = \frac{1}{\beta_{j+1}}\,\widetilde{s}_{j+1} && \text{(normalization step)}
\end{aligned}$$

until $\alpha_j = 0$ or $\beta_{j+1} = 0$, or the dimension n of the problem is reached. Assume that the process does not terminate before the step k. Then the left bidiagonalization vectors $s_1, \ldots, s_k$ represent an orthonormal basis of the Krylov subspace

$$\mathcal{K}_k(AA^T, b) \equiv \operatorname{span}\{b, AA^T b, \ldots, (AA^T)^{k-1} b\},$$

and the right bidiagonalization vectors $w_1, \ldots, w_k$ represent an orthonormal basis of the Krylov subspace

$$\mathcal{K}_k(A^T A, A^T b) \equiv \operatorname{span}\{A^T b, A^T A A^T b, \ldots, (A^T A)^{k-1} A^T b\}.$$

Denote

$$L_k \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix} \in \mathbb{R}^{k\times k} \tag{2.1}$$

the bidiagonal matrix of the normalization coefficients, representing a projection (restriction) of the operator A onto the above defined k-dimensional Krylov subspaces,

$$L_k = [s_1, \ldots, s_k]^T A\,[w_1, \ldots, w_k];$$

see [4], [15]. Let $p_1^{(k)}$ be the left singular vector corresponding to the smallest singular value of L_k.[1]

In [11] it was described how *white noise* from the right-hand side b propagates in the Golub–Kahan iterative bidiagonalization, particularly in the left bidiagonalization vectors; see also [10], [12]. While the starting vector s_1 is smooth (since it is dominated by the scaled b^{exact}), during the bidiagonalization process, as k increases, the left bidiagonalization vectors s_k become more and more dominated by the high-frequency part of propagated noise b^{noise}. This is caused by projecting out the low-frequency components (arising mostly from b^{exact} and partly also from the low-frequency part of b^{noise}) in order to achieve orthogonality among the bidiagonalization vectors. The iteration where the most high-frequency dominated vector is obtained is called the *noise revealing* iteration and is denoted by k_{noise}. After this iteration, a part of noise is projected out resulting in a smoother left bidiagonalization vector. Analysis of this phenomenon in [11] allowed to derive two quantities estimating the noise level: The *cumulative (amplification) ratio*

$$\varphi_k = \prod_{j=1}^{k} \frac{\alpha_j}{\beta_{j+1}} \tag{2.2}$$

and *the size of the first entry* of $p_1^{(k)}$, i.e.,

$$|(p_1^{(k)}, e_1)|, \tag{2.3}$$

where $(\cdot\,,\cdot)$ denotes the standard inner product and $e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^k$. It was proved that (2.2) and (2.3) both (on average) decrease until k_{noise}. After this iteration, the cumulative ratio increases while the size of the first entry of $p_1^{(k)}$ begins to almost stagnate. This allows to detect the iteration k_{noise} in which the best noise level approximation is obtained; see [11] and also [17]. Note that both estimators are relatively cheap to compute. Since noise usually propagates rapidly, k_{noise} is very small in comparison to n. The bidiagonalization coefficients α_j, β_{j+1} are directly available, computation of the singular vector $p_1^{(k)}$ for a small bidiagonal matrix L_k can be performed efficiently.

We now use the example from [11] to illustrate the behavior of both estimators on the problem `shaw` from the Regularization Toolbox [5] in MATLAB. This problem represents a one-dimensional image restoration model obtained as a quadrature

[1] Note that we use the notation introduced in [11].

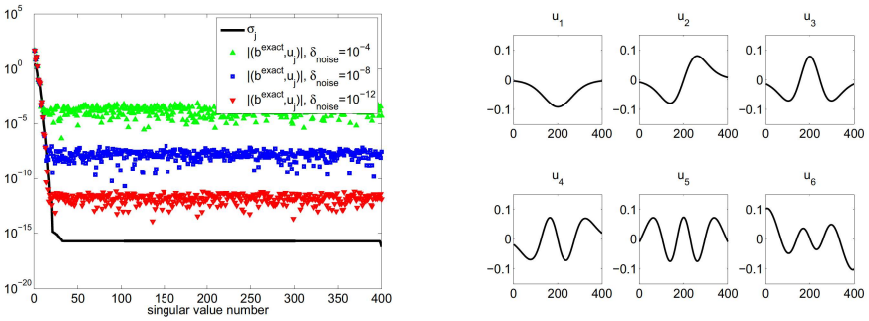


FIG. 2.1. *Illustration of the violation of the discrete Picard condition for the problem* `shaw(400)` *with white noise and the noise levels* $\delta_{\text{noise}} = 10^{-4}$, 10^{-8}, *and* 10^{-12} *(left). Increasing frequencies in the first six left singular vectors of* A *for the problem* `shaw(400)` *(right).*

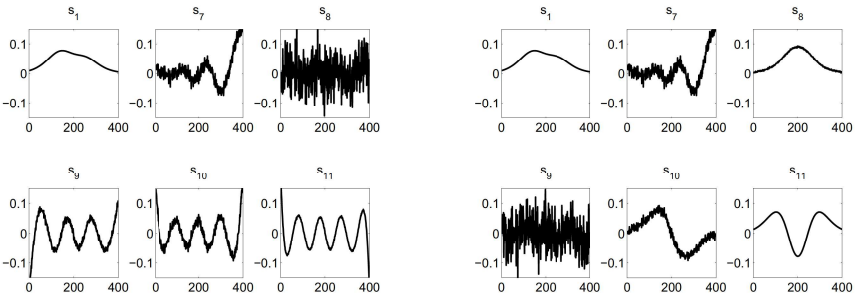


FIG. 2.2. *Comparison of several left bidiagonalization vectors* s_k *computed by the Golub–Kahan iterative bidiagonalization implemented with (left) and without (right) reorthogonalization, for the problem* `shaw(400)` *with white noise and the noise level* $\delta_{\text{noise}} = 10^{-4}$.

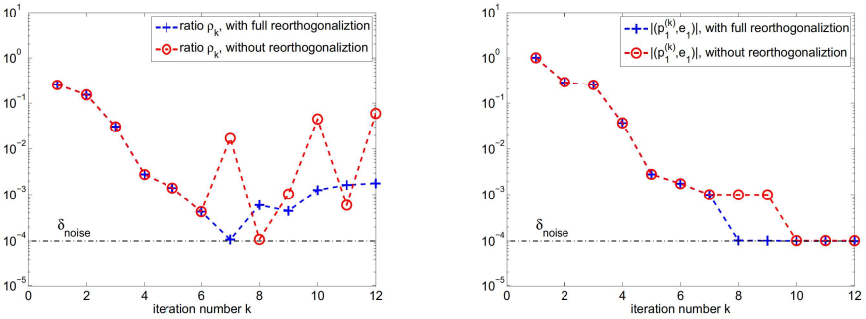


FIG. 2.3. *Estimates obtained by cumulative amplification ratios (left), and sizes of the first entry of* $p_1^{(k)}$ *(right), for the problem* `shaw(400)` *with white noise and the noise level* $\delta_{\text{noise}} = 10^{-4}$. *Computations were performed with and without reorthogonalization.*

discretization of a first kind Fredholm integral equation on the integration intervals $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Here, the smoothing kernel is given by

$$K(s,t) = \Big(\cos(s) + \cos(t)\Big)^2 \left(\frac{\sin(u)}{u}\right)^2, \quad \text{where} \quad u = \pi\Big(\sin(s) + \sin(t)\Big);$$

see [16] for the description of the model. The linear problem with the size $n = 400$ is

contaminated by white noise generated by `randn(400,1)` rescaled to obtain a particular noise level δ_{noise}. Figure 2.1 (left) shows the sizes of the projections (b, u_j), where u_j are the left singular vectors of the matrix A corresponding to the singular values ordered in the nonincreasing order. We see how the presence of noise of various noise level results in the violation of the discrete Picard condition for the subspaces corresponding to smaller singular values. Figure 2.1 (right) presents increasing frequencies in the left singular vectors of A.

Figure 2.2 illustrates how white noise reveals in the left bidiagonalization vectors for $\delta_{\text{noise}} = 10^{-4}$. The vectors in the left part are computed with full double reorthogonalization in the Golub–Kahan iterative bidiagonalization in order to simulate exact arithmetic. Clearly, the frequencies increase before they become maximal in s_8, computed in the iteration $k_{\text{noise}} = 7$; see the algorithm above. The right part shows the delay in noise revealing caused by reappearance of a smooth vector, as a result of the loss of orthogonality in the bidiagonalization implemented without reorthogonalization. However, the effect is still present and $k_{\text{noise}} = 8$. Figure 2.3 compares estimates obtained by cumulative amplification ratios (left) and by sizes of the first entry of $p_1^{(k)}$ (right). We see that both estimators give very accurate and comparable results for computation with as well as without reorthogonalization. It is worth noting that oscillations in the cumulative ratio computed without reorthogonalization can cause difficulties in automatic detection of k_{noise}. Thus, in the following we restrict ourselves only to the estimate (2.3). Analysis of the methods detecting the point of stagnation is out of the scope of this paper; see [17] for some ideas.

3. Performance for 2D image deblurring problems. Robustness of the estimator (2.3) is studied on a sharp testing picture X of size 167×250 pixels, i.e., $n = 41\,750$; see Figure 3.1. The experiments are performed in MATLAB, with the use of functions from the Image Processing Toolbox.

3.1. Spatially invariant blur. First we consider a standard spatially invariant blurring model, where blurring of each individual pixel in X is characterized by a given point-spread-function (PSF); see [8, Chapter 3]. Presented results include models for a Gaussian blur, motion blur, and disc blur. Using the function `fspecial`, we construct two PSFs with smaller and larger support for each type of blur, giving in total six testing PSFs; see Figure 3.2. Note that since we only need to perform matrix-vector multiplications, we do not form the corresponding blurring matrix A explicitly. The blurred images B are computed by the 2D convolution using the function `conv2` with the parameter `valid`, i.e., only the part computed without the zero-padded edges is returned. Multiplication by the matrix A in the bidiagonalization is performed by the function `conv2` with the parameter `same` representing zero boundary conditions. For testing purposes, the image B is contaminated by noise using the function `imnoise` with four different parameter settings. We consider two types of noise: white noise with Gaussian distribution (parameter `gaussian`), and uniformly distributed data-correlated noise (parameter `speckle`). Variances $\sigma^2 = 10^{-2}$ and $\sigma^2 = 5 \cdot 10^{-6}$ give two different noise levels δ_{noise} for each type of noise.

Figure 3.3 provides similar information as Figure 2.1, here for the matrix A corresponding to the larger Gaussian PSF and white noise. Again we see the violation of the discrete Picard condition. The so-called left singular images (reshaped left singular vectors) of the blurring matrix A tend to be dominated by higher frequencies, i.e., more oscillations appear in both vertical and horizontal directions, as k increases; see [8] for details.

FIG. 3.1. *Sharp testing image X of size 167×250 pixels.*

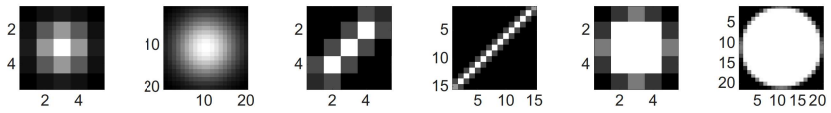


FIG. 3.2. *Considered PSFs defining the blurring matrix A. From the left: two Gaussian, two motion and two disc PSFs.*

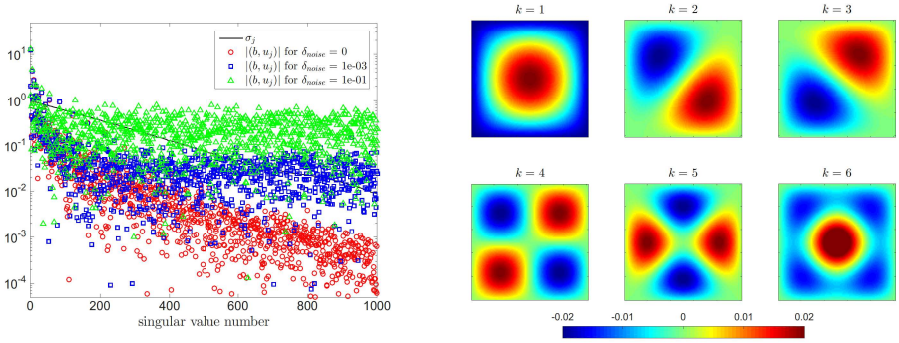


FIG. 3.3. *Illustration of violation of the discrete Picard condition for the 2D image deblurring problem with larger Gaussian blur and white noise of various noise levels (left). Increasing frequencies in the first six left singular vectors of the corresponding blurring matrix A printed as 2D images (right).*

Figure 3.4 shows blurred noisy images together with the corresponding noise level estimates obtained for models with the smaller Gaussian blur, for the four above described noise settings. Bidiagonalization with and without reorthogonalization is used. First, we observe that the overall behavior of the estimator does not significantly depend on the reorthogonalization, except for the fact that for lower noise levels the delay in noise revealing results in the increase of the computational cost, since more iteration steps are required to obtain a reasonable estimate (note the different scaling of the x axis in the second and fourth row). This problem is not present in experiments with more realistic higher noise level (the first and third row), where noise propagates quickly. This is a positive message since for larger images, reorthogonalization cannot be performed because of its enormous computational cost and memory requirements. Furthermore, we see that the expected stagnation in the estimator allowing to detect the noise revealing iteration is more significant for higher noise levels. Figure 3.5 is the counterpart of Figure 3.4 for larger Gaussian blur. We observe clear stagnation

in all curves. The behavior of the estimator for higher noise levels is generally very similar to Figure 3.4, however we see that in Figure 3.5 the lower noise levels are overestimated.

Summarizing, the best results are obtained for large blur and large noise levels, shown in the first and third row of Figure 3.5 making the estimator more successful on complicated problems. However, for noise below a certain level problems appear. For smaller amount of blur the noise revealing iteration can not be perfectly detected, while for larger amount of blur the noise level is overestimated. This is more significant in experiments with data-correlated noise, which is not surprising as the estimator is based on revealing of the *high-frequency part of noise* in the vectors s_k, thus it does not take into account the smooth part of noise. For white noise, this does not represent a complication. However, even though data-correlated noise is white-noise like, its behavior partly resembles behavior of the smooth noise-free data in b.

Figure 3.6 gives noise level estimates for two variants of the motion and disc blur (specified on the top) with four different noise settings (specified on the left). All results were obtained by the Golub–Kahan iterative bidiagonalization implemented without reorthogonalization. We see similar results as in experiments with the Gaussian blur. Consequently, the blur type has generally minor impact on the performance of the estimator.

3.2. Spatially variant blur. In addition to spatially invariant blur, we investigate the behavior of the estimator (2.3) for a special type of spatially variant blur: a rotational blur recently studied in the context of image deblurring in [9]. Consider a sharp image represented by the central part of size 167×167 of the image from Figure 3.1 in order to avoid the large black areas appearing at the edges when rotating the whole rectangular image. The code to construct the rotational blurring operator has been provided by Per Christian Hansen, and it is identical to the code used in [9]. We consider three different blurs: rotation by 10°, rotation by 20°, and tilt by 10°. All the resulting blurred images are corrupted by additive white noise with two different variances $\sigma^2 = 10^{-3}$ and $\sigma^2 = 5 \cdot 10^{-6}$. The noisy images together with noise level estimates computed by the Golub–Kahan iterative bidiagonalization without reorthogonalization for all settings are shown in Figure 3.7.

The results are very similar to results for the spatially invariant blurring. The estimates for large noise level are accurate (left). Especially in case of strong blurring (rotation by 20°), the curve stagnates very close to the actual noise level. For the smaller noise level, the stagnation is not so significant.

4. Conclusions. Presented paper has studied performance of the noise level estimator proposed in [11], which is based on the iterative Golub–Kahan bidiagonalization, on image deblurring problems. Implementations with and without reorthogonalization have been compared. We have demonstrated that reorthogonalization does not improve the quality of the estimate, although for small noise levels we would need more iterations to obtain estimate of the same accuracy as by the algorithm with reorthogonalization. We have shown that the performance of the estimator does not significantly depend on the particular type of blur but it is generally more successful on problems with higher noise levels. For smaller noise levels, the expected stagnation of the estimator has been rather slow, making the detection of the noise revealing iteration (where the best estimate should be obtained) complicated. For data-correlated noise of lower noise level, the estimator has not been reliable, as it underestimated some noise levels while it overestimated the others. Further analysis of the observed

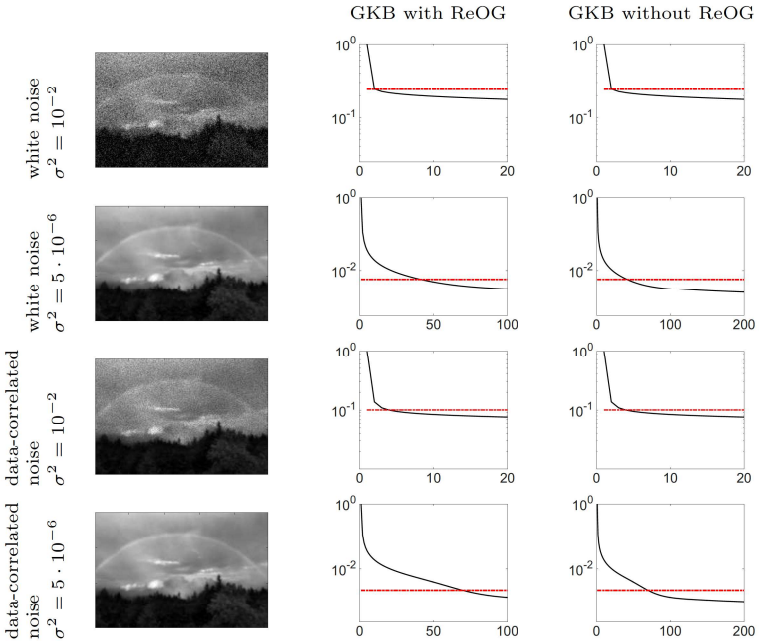


FIG. 3.4. *Blurred noisy images and the corresponding noise level estimates for smaller Gaussian blur and four considered noise settings (specified on the left). Computed by the Golub–Kahan bidiagonalization (GKB) with and without reortogonalization (ReOG) (specified on the top). Red line represents the exact noise level.*

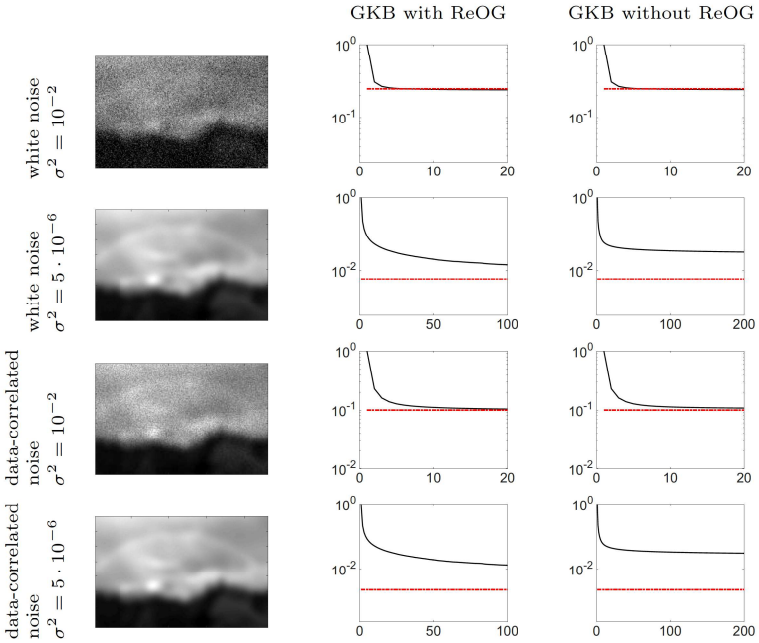


FIG. 3.5. *Results similar as in Figure 3.4 computed for the model with the larger Gaussian blur.*

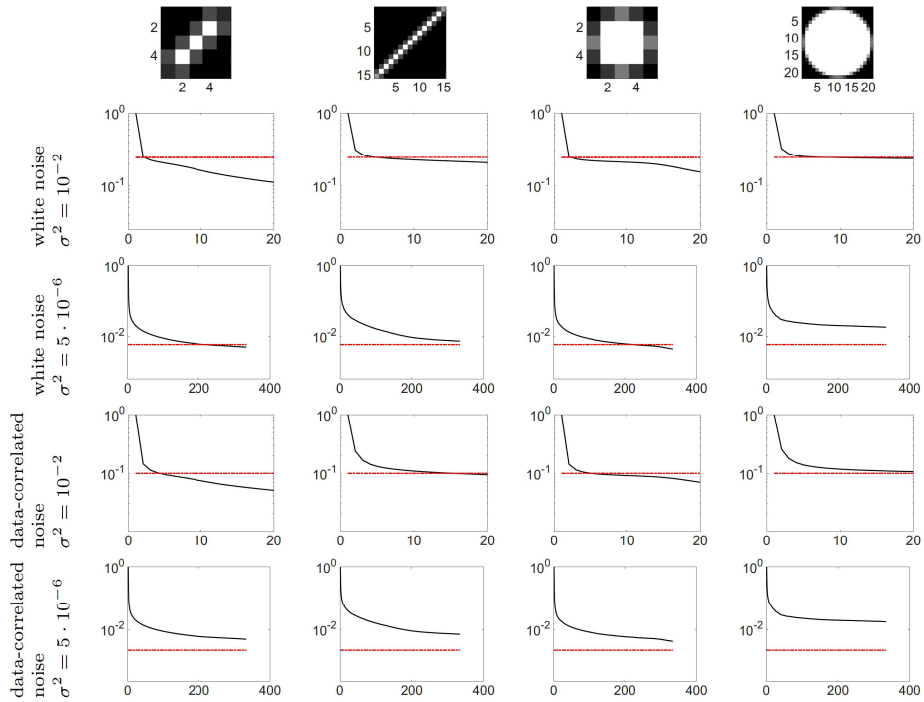


FIG. 3.6. *Noise level estimates for models with the motion and disc blur (specified on the top) and four considered noise settings (specified on the left), computed without reortogonalization.*

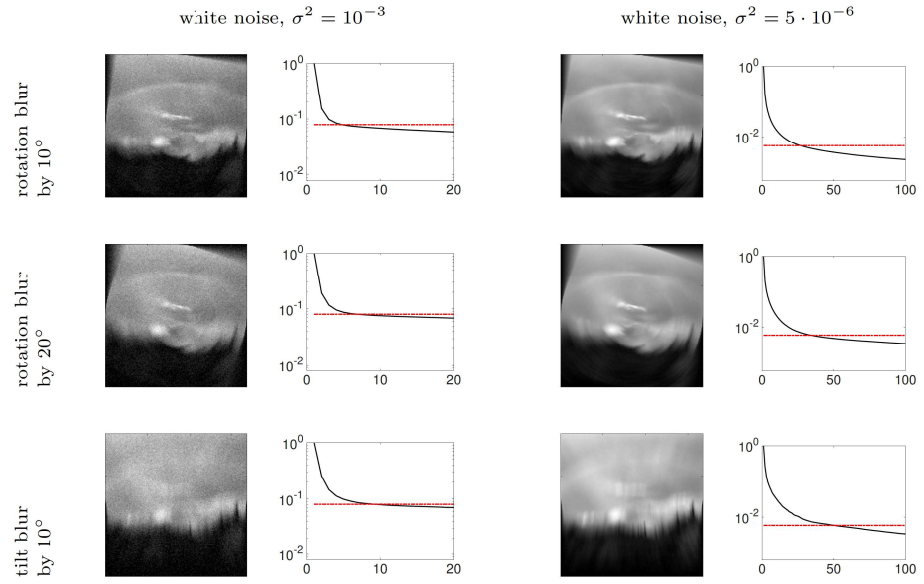


FIG. 3.7. *Noise level estimates for models with the rotational blur and white noise with two different variances (specified on the top), computed without reortogonalization.*

behavior and related issues is out of the scope of this paper and will be presented elsewhere.

REFERENCES

[1] P. Blomgren and T. F. Chan, *Modular solvers for constrained image restoration problems using the discrepancy principle*, Numer. Linear Algebra Appl., 9 (2002), pp. 348–358.

[2] L. Desbat and D. Girard, *The "minimum reconstruction error" choice of regularization parameters: Some more efficient methods and their application to deconvolution problems*, SIAM J. Sci. Comput., 16 (1995), pp. 1387–1403.

[3] S. Gazzola, P. Novati, and M. R. Russo, *On Krylov projection methods and Tikhonov regularization*, ETNA, 44 (2015), pp. 83–123.

[4] G. H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., Ser. B 2 (1965), pp. 205 224.

[5] P. C. Hansen, *Regularization Tools Version 4.1 (for MATLAB Version 7.3). A MATLAB package for analysis and solution of discrete ill-posed problems* (available at `http://www.imm.dtu.dk/~pcha/Regutools`).

[6] ———, *Rank-Deficient and Discrete Ill-Posed Problems, Numerical Aspects of Linear Inversion*, SIAM Publications, Philadelphia, PA, 1998.

[7] P. C. Hansen and T. K. Jensen, *Noise propagation in regularizing iterations for image deblurring*, ETNA, 31 (2008), pp. 204–220.

[8] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM Publications, Philadelphia, PA, 2006.

[9] P. C. Hansen, J. G. Nagy, and K. Tickos, *Rotational image deblurring with sparse matrices*. BIT Numerical Mathematics, 54 (2013), pp. 649–671.

[10] I. Hnětynková, M. Kubínová, and M. Plešinger, *On noise propagation in residuals of Krylov subspace iterative regularization methods*, submitted.

[11] I. Hnětynková, M. Plešinger, and Z. Strakoš, *The regularizing effect of the Golub–Kahan iterative bidiagonalization and revealing the noise level in the data*, BIT Numerical Mathematics, 49 (2009), pp. 669-696.

[12] M. Michenková, *Regularization techniques based on the least squares method*, Diploma thesis, Charles University in Prague, 2013.

[13] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl., 7 (1966), pp. 414–417.

[14] F. Natterer, *The Mathematics of Computerized Tomography*, John Wiley & Sons and Teubner, Stuttgart, 1986.

[15] C. C. Paige, *Bidiagonalization of matrices and solution of linear equations*, SIAM J. Numer. Anal., 11 (1974), pp. 197–209.

[16] C. B. Shaw, Jr., *Improvements of the resolution of an instrument by numerical solution of an integral equation*, J. Math. Anal. Appl., 37 (1972), pp. 83–112.

[17] K. Vasilík, *Linear algebraic modeling of problems with noisy data*, Diploma thesis, Charles University in Prague, 2011.

3.2 Influence of the matrix shape

In Hnětynková et al. [2009], the noise-level estimator was considered for square nonsingular matrices only. In image deblurring applications considered in the previous part, the matrices A representing the discretization of the blurring operator are naturally square, as long as the original sharp image and its blurred counterpart have the same size. There are however many discrete inverse problems, such as those arising in computerized tomography, see, e.g., Natterer [1986] or seismic tomography, see, e.g., Sheriff and Geldart [1995], whose system matrix is rectangular, with either fewer or more columns than rows. In the following we show how the performance of the noise-level estimator derived in Hnětynková et al. [2009] is influenced by the shape of the matrix. In order to do that, it is crucial to understand how the matrix shape affects the underlying distribution function.

Let $A \in \mathbb{R}^{m\times n}$ and let

$$A = U\Sigma V^T$$

be its singular value decomposition, with $U^{-1} = U^T$ and $V^{-1} = V^T$, and $\Sigma \in \mathbb{R}^{m\times n}$ being a diagonal matrix with the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(m,n)} > 0$ on its diagonal. The quality of the considered estimator is determined by the behavior of the underlying distribution function ω with the nodes defined by the m eigenvalues of AA^T, i.e., the diagonal entries of $\Sigma\Sigma^T$, and the corresponding weights $|u_1^T b|^2/\|b\|^2, \ldots, |u_m^T b|^2/\|b\|^2$. Since the right-hand side b is a sum of the noise-free data Ax and noise η, i.e.,

$$b = Ax + \eta,$$

depending on the noise level, the weights become eventually dominated by the noise vector η, see [Hnětynková et al., 2009, sec. 4.2]. Recall that

$$\delta^2_{\text{noise}} \equiv \frac{\|\eta\|^2}{\|Ax\|^2} = \sum_{i=1}^{m} \frac{|u_i^T\eta|^2}{\|Ax\|^2}$$

and

$$\omega(t) = \sum_{i=k}^{m} \frac{|u_i^T b|^2}{\|b\|^2} = \sum_{i=k}^{m} \frac{|u_i^T Ax + u_i^T\eta|^2}{\|Ax+\eta\|^2}, \quad \text{for} \quad t \in [\sigma_k^2, \sigma_{k-1}^2).$$

We now show how the shape of the distribution function is influenced by the dimension of the matrix A.

Let us fix the true solution $x \in \mathbb{R}^n$. If the matrix A has more rows than columns, i.e., $m > n$, then the multiple node 0 (corresponding to the smallest $m-n$ eigenvalues of AA^T) has the weight $\sum_{i=n+1}^{m}(u_i^T\eta)^2/\|b\|^2$, i.e.,

$$\omega(t) = \sum_{i=n+1}^{m} \frac{|u_i^T\eta|^2}{\|Ax+\eta\|^2}, \quad \text{for} \quad t \in [0, \sigma_n^2),$$

resulting in a flatter distribution function in the subintervals dominated by noise and therefore more distinct stagnation around the squared noise level δ^2_{noise}. If on

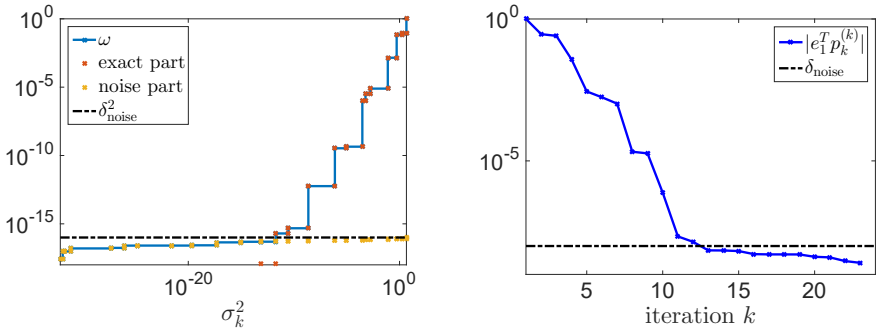
the other side A has more columns than rows, i.e., $m < n$, then the distribution function ω only has m points of increase, from which we conclude that ω is overall steeper than for the overdetermined or square system. In many cases, such as when solving discretized Fredholm equations, the behavior of $|u_i^T Ax|^2$, $i = 1, \ldots, \min(m, n)$, is rather independent of the shape of the matrix. This is because the projections of Ax to the directions u_i, representing the approximation of the left singular functions, see Hansen [1988], are not influenced by taking more measurements since Ax tends to be very smooth. Therefore, the stage where $|u_i^T Ax + u_i^T \eta|^2$ is dominated by $|u_i^T Ax|^2$ remains unchanged and the steeper increase is localized in the part where $|u_i^T \eta|^2$ dominates, leading to less distinct stagnation around the squared noise level.

Since the construction of the distribution function is unfeasible in practical computations, it is approximated through entities computed in the Golub-Kahan bidiagonalization applied to matrix A and starting vector b. Let L_k be the lower-bidiagonal matrix of the normalization coefficients generated in the Golub-Kahan iterative bidiagonalization and

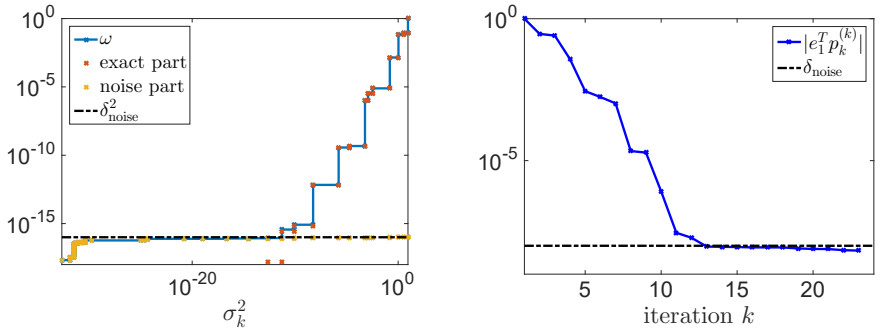
$$L_k = P^{(k)} \Theta^{(k)} (Q^{(k)})^T$$

its singular value decomposition with the singular values $\theta_1^{(k)}, \ldots, \theta_k^{(k)}$ on its diagonal ordered in the nonincreasing order. Then in each step k the distribution function ω is approximated by the distribution function $\omega^{(k)}$ with the k nodes $(\theta_j^{(k)})^2$ and the corresponding weights $(e_1^T p_j^{(k)})^2$. The stagnation of $(e_1^T p_k^{(k)})^2$ indicates that δ_{noise}^2 has been reached, allowing to estimate the noise level, see the previous section or [Hnětynková et al., 2009, sec. 4.1].

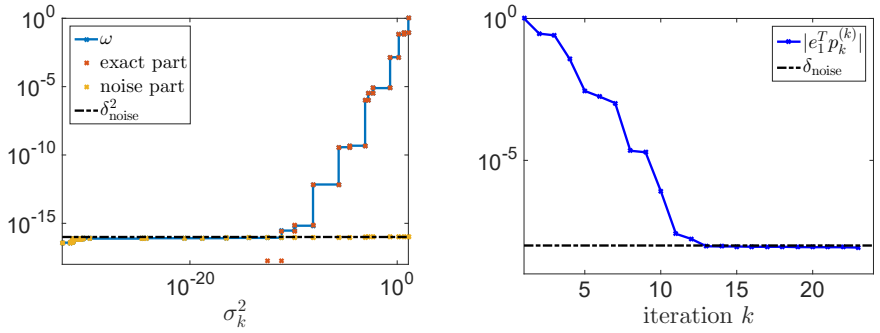
For illustration we consider the problem `shaw` from Hansen [1994], representing a discretization of Fredholm integral equation of the first kind with a square-integrable kernel. We modified the MATLAB function to be able to handle different dimensions of the output and input to generate rectangular matrices. We assume $n = 48$, which is the number of discrete points of the continuous true solution, and $m = 24, 48, 64$, corresponding to the number of measurements taken. This setting leads to an under-determined, a square, and an over-determined system of linear equations. Gaussian noise with $\delta_{\text{noise}} = 10^{-8}$ is added to each of the right-hand sides. For each of the settings we show in the left part of Figure 3.1 the corresponding distribution function ω. On the right, we plot the size of $e_1^T p_k^{(k)}$ against the iteration k. In each case we only perform 24 steps of the Golub-Kahan bidiagonalization, which is the maximum number of steps for the under-determined case with $m = 24$. We perform full double reorthogonalization to simulate exact arithmetic, see also Section 2.2. In Figure 3.1 we indeed observe that the part corresponding to the exact data, represented by $\sum_{i=k}^{m} \frac{|u_i^T Ax|^2}{\|Ax+\eta\|^2}$ plotted against σ_k^2, of the distribution function ω is not much affected by the number of measurements. The part corresponding to noise, represented by $\sum_{i=k}^{m} \frac{|u_i^T \eta|^2}{\|Ax+\eta\|^2}$ plotted against σ_k^2, becomes steeper if we decrease the number of columns, making the stagnation at the noise level less significant. Noise-level estimation from the approximation of the distribution function ω computed from the bidiagonalization is then, due to less significant corner in the curve, more difficult, see Figure 3.1



(a) $m = 24$



(b) $m = 48$



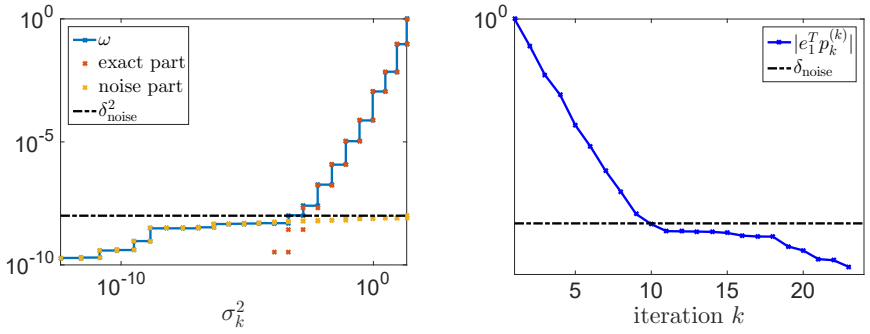
(c) $m = 64$

Figure 3.1: Noise level estimation for the problem `shaw` of size $m \times 48$, $m = 24, 48, 64$, with Gaussian white noise with the noise level $\delta_{\text{noise}} = 10^{-8}$. Left: distribution function ω together with the part corresponding to the exact data $\sum_{i=k}^{m} \frac{|u_i^T Ax|^2}{\|Ax+\eta\|^2}$ and the part corresponding to noise $\sum_{i=k}^{m} \frac{|u_i^T \eta|^2}{\|Ax+\eta\|^2}$ plotted both against σ_k^2. Right: Bidiagonalization-based noise-level estimator, first 24 iterations. Performance of the estimator, i.e., stagnation around the noise level, deteriorates if the number of measurements m is too small.

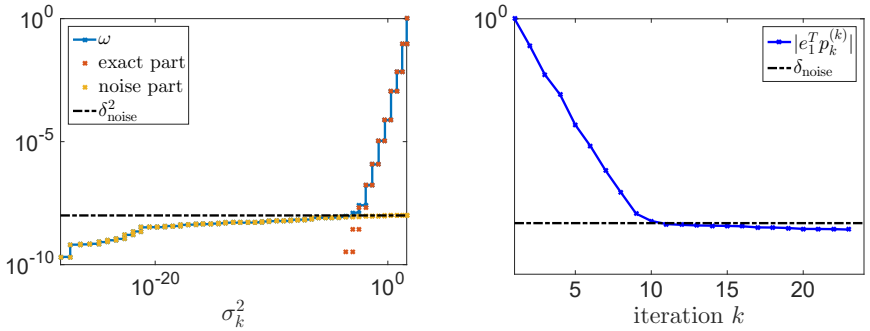
right. Analogous experiment for problem `gravity` adopted from Hansen [1994], modified to generate rectangular matrices, is shown in Figure 3.2.

The shape of the matrix only plays important role in the cases when $|e_1^T p_k^{(k)}|$ reaches the noise level relatively late with respect to the size of the problem. If the noise level is reached in early iterations, the shape of the matrix has very little influence on the significance of stagnation, as it has very little influence

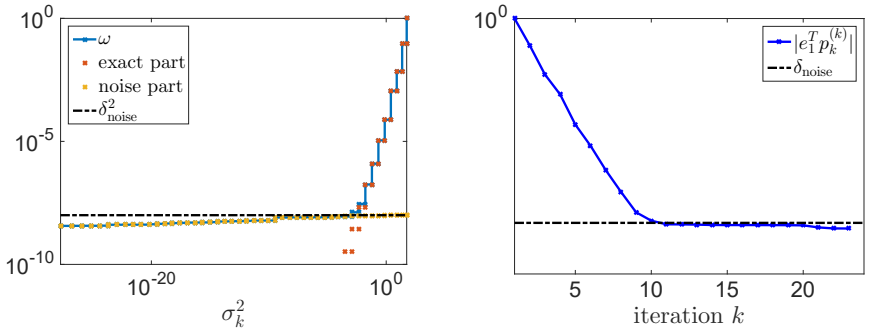
on the shape of the distribution function itself. This is because $\sum_{i=k}^{m} \frac{|u_i^T \eta|^2}{\|Ax+\eta\|^2}$ plotted against σ_k^2 in logarithmic scale is almost constant for $k \ll \min(m,n)$. We demonstrate this in Figure 3.3, where we consider the same problem as in Figure 3.1, except that we take ten times more discretization points both in the source and the data. Note that the estimator has practical importance only for cases when the noise level is reached early, i.e., relatively few iteration of the iterative bidiagonalization need to be performed, in which case the shape of the matrix plays a minor role.



(a) $m = 24$

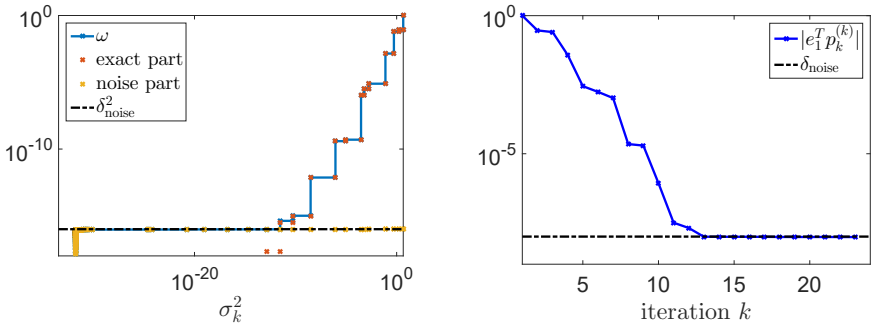


(b) $m = 48$

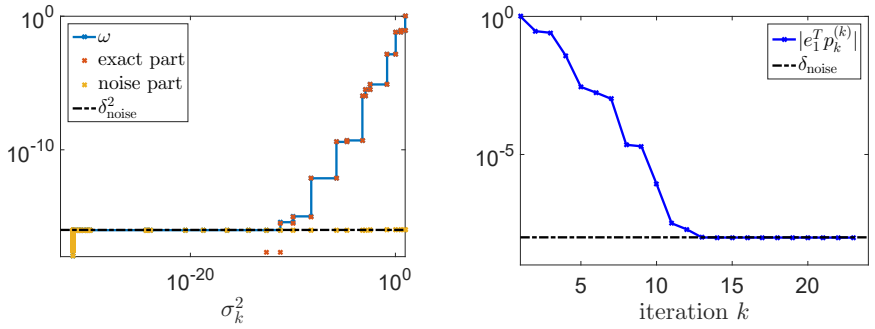


(c) $m = 64$

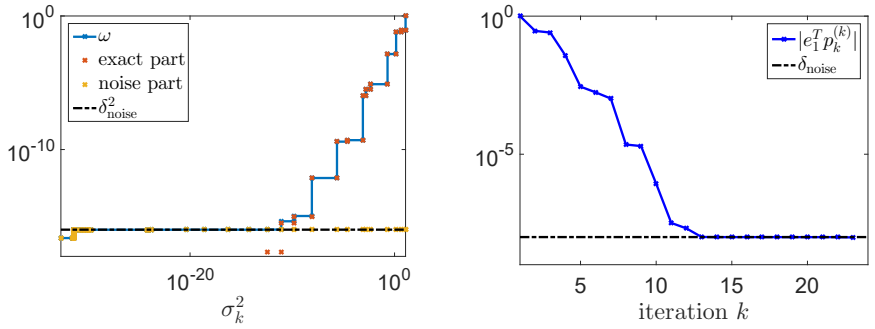
Figure 3.2: Noise level estimation for the problem `gravity` of size $m \times 48$, $m = 24, 48, 64$, with Gaussian white noise with the noise level $\delta_{\text{noise}} = 10^{-4}$. See the description in Figure 3.1 for further details. Performance of the estimator, i.e., stagnation around the noise level, deteriorates if the number of measurements m is too small.



(a) $m = 240$



(b) $m = 480$



(c) $m = 640$

Figure 3.3: Noise level estimation for the *larger* problem `shaw` of size $m \times 480$, $m = 240, 480, 640$ and the other setting same as in Figure 3.1. The noise level is reached in early iterations (~ 15) with respect to the size of the matrix, therefore the matrix shape has little influence.

Bibliography

P. C. Hansen. Computation of the singular value expansion. *Computing*, 40(3): 185–199, 1988.

P. C. Hansen. Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6(1-2):1–35, 1994.

I. Hnětynková, M. Plešinger, and Z. Strakoš. The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data. *BIT*, 49(4):669–696, 2009.

I. Hnětynková, M. Kubínová, and M. Plešinger. Notes on performance of bidiagonalization-based noise level estimator in image deblurring. In A. Handlovičová, editor, *Proceedings of the Conference Algoritmy*, pages 333–342. Slovak University of Technology in Bratislava, Publishing House of STU, 2016.

F. Natterer. *The mathematics of computerized tomography*. Vieweg+Teubner Verlag, 1986.

R. E. Sheriff and L. P. Geldart. *Exploration seismology*. Cambridge university press, 1995.

4. Delay of approximation properties of Krylov subspace methods in finite-precision arithmetic

Krylov subspace methods for solving systems of linear equations or discrete inverse problems generally rely on construction of a well-conditioned, typically orthonormal basis of the corresponding Krylov subspace. Methods constructing these bases using short recurrences, i.e., without explicit orthogonalization, are appealing both for their computational efficiency and low storage requirements. In finite-precision computations, short recurrences however often lead to the loss of global orthogonality or even to linear dependence of the computed vectors, which subsequently causes the delay of convergence of the related method. This was observed already by Lanczos [1950]. Surprisingly, the loss of orthogonality occurring in practical computations does not lead to a complete deterioration of the approximation properties of the methods. Over the years, many researchers contributed to understanding of why this is the case; see, e.g., Greenbaum [1989]; Paige [1980], or Meurant and Strakoš [2006] for the overview. While the essence of the relationship between the exact computation and the finite-precision computation *with the same input data* is to some extent understood, its quantitative interpretation is still missing.

Due to the loss of orthogonality and loss of the linear independence, the computed vectors efficiently span subspace of smaller dimension, comparing the first k computed vectors from finite-precision arithmetic with the first k exact basis vectors is therefore pointless. On the other side, comparing the computed vectors with the exact vectors from some earlier iteration $l < k$, where l corresponds to the number of numerically linearly independent vectors from those computed in the k-th iteration, might theoretically be possible. This approach was applied, e.g., in [Liesen and Strakoš, 2013, sec. 5.9.1] and [Gergelits, 2013, chap. 3], and the results presented there suggest that the finite-precision computation is to some extent only a delayed version of its exact counterpart applied to the same data, here in the context of the energy norm of the error in the conjugate gradient method. We are interested whether and how this association through a delay might be used to link the exact and the finite-precision Krylov subspace computations with the same input data in a broader context, mainly in the sense of:

- the convergence of the resulting methods;
- the 'basis' vectors generated sequentially in each iteration;
- the solution and residual vectors;

- other entities, such as the Ritz values and the Ritz vectors.

In this work, we focus on entities whose size does not decay monotonically during the computation, such as for example the residuals of Galerkin methods. Section 4.1 includes a proceedings contribution showing that such entities do not allow direct association between the finite-precision and exact computation, and that a more sophisticated strategy based on aggregation over the intermediate iterations must be employed. The contribution also contains some ideas about how to find, for a given finite-precision iteration k, the associated iteration l in exact arithmetic.

In section 4.2, we discuss the relationship between the computed Ritz vectors and the exact ones and show some preliminary results in this direction. We acknowledge the contribution of Tomáš Gergelits to Section 4.2.

4.1 Contribution in Proceedings of HPCSE conference

This section contains the contribution Gergelits et al. [2018]. Reprinted by permission from Springer Nature: *High Performance Computing in Science and Engineering. HPCSE 2017. Lecture Notes in Computer Science*, Gergelits, T., Hnětynková, I. & Kubínová, M.: Relating computed and exact entities in methods based on Lanczos tridiagonalization, copyright (2018), (doi:10.1007/978-3-319-97136-0_6).

Relating Computed and Exact Entities in Methods Based on Lanczos Tridiagonalization

Tomáš Gergelits[1,2], Iveta Hnětynková[1], and Marie Kubínová[1,2(✉)]

[1] Faculty of Mathematics and Physics, Charles University,
121 16 Prague, Czech Republic
{gergelits,hnetynko,kubinova}@karlin.mff.cuni.cz
[2] Institute of Computer Science, The Czech Academy of Sciences,
182 07 Prague, Czech Republic

Abstract. Krylov subspace methods based on short recurrences such as CGL or MINRES represent an attractive way of solving large and sparse systems of linear algebraic equations. Loss of orthogonality in the underlying Lanczos process delays significantly their convergence in finite-precision computation, whose connection to exact computation is still not fully understood. In this paper, we exploit the idea of simultaneous comparison of finite-precision and exact computations for CGL and MINRES, by taking advantage of their relationship valid also in finite-precision arithmetic. In particular, we show that finite-precision CGL residuals and Lanczos vectors have to be aggregated over the intermediate iterations to form a counterpart to vectors from the exact computation. Influence of stagnation in exact MINRES computation is also discussed. Obtained results are supported by numerical experiments.

Keywords: Krylov subspace · CGL · MINRES
Finite-precision computations · Loss of orthogonality
Delay of convergence · Lanczos vectors

1 Introduction

Large and sparse linear algebraic problems of a general form

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n},\ b \in \mathbb{R}^n,$$

can be often solved efficiently by Krylov subspace methods. Many of these rely mathematically on computation of an orthonormal basis of the Krylov subspaces

$$\mathcal{K}_k(A, r_0) \equiv \mathrm{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}, \quad k = 1, 2, \ldots, \tag{1}$$

where $r_0 = b - Ax_0$, with x_0 being the initial approximation. For a symmetric A, such basis can be efficiently computed by *short recurrences*, represented by

© Springer International Publishing AG, part of Springer Nature 2018
T. Kozubek et al. (Eds.): HPCSE 2017, LNCS 11087, pp. 73–87, 2018.
https://doi.org/10.1007/978-3-319-97136-0_6

the Lanczos algorithm [11]. However, in finite-precision arithmetic, the global orthogonality and subsequently also the linear independence of the computed Lanczos vectors $\bar{v}_j$ is usually quickly lost, and the subspaces spanned by $\bar{v}_j$ are not Krylov subspaces defined by the input data. As a result, the convergence of methods such as the Conjugate gradients implemented via the Lanczos algorithm (CGL) [9,11] or the Minimal residual method (MINRES) [17], is *significantly delayed*, and the computed entities, including approximate solutions or residuals, can deviate substantially from their mathematical counterparts; see [6]. For some problems, the computation may not be affected by this delay, for example because the desired accuracy of the approximate solution is reached before the severe loss of orthogonality emerges (e.g., when efficient preconditioning can be used). However, short recurrences in principle cannot guarantee the linear independence of the computed vectors when rounding errors are present. Various techniques for preserving orthogonality, such as the full or selective reorthogonalization (see, e.g., [21,22] or [13, Sect. 4.5]) have been developed, but for large scale problems, reorthogonalization in the Lanczos algorithm is typically unaffordable since it heavily increases computational time and storage requirements.

The first significant step in explaining the behavior of the Lanczos algorithm in finite-precision arithmetic was made in [15,16]. It was proved that the loss of orthogonality among the computed Lanczos vectors is possible only in the directions of eigenvectors of the matrix A (more specifically, in the directions of Ritz vectors associated with converged Ritz values). Another fundamental step was done in [6,7] showing that the behavior in the first k steps of the finite-precision Lanczos computations is identical to the behavior of exact Lanczos computations applied to a possibly larger matrix $\widehat{A}(k)$, whose eigenvalues lie within tiny intervals around the eigenvalues of A; see also [13] for an overview. In [19] the finite-precision Lanczos process in step k is described via the exact Lanczos process applied on augmented system containing both the matrix A and the currently computed tridiagonal Jacobi matrix. Sensitivity of Krylov subspace to small perturbations of the input data was studied in [1,10,20]. However, these results assume linear independence of the computed Lanczos vectors, which is often quickly lost.

Despite the wide attention, the properties of the methods based on the Lanczos process are in finite-precision computations still not fully understood. In particular, it is not clear how the subspaces generated by the computed Lanczos vectors differ from the exact Krylov subspaces, or how the computed approximation or residual vectors resemble their counterparts from exact computation with the *same matrix and starting vector*. The approaches in [6,7,19] do not allow direct comparison of the solution, residual, or Lanczos vectors, since they involve extended or augmented matrices.

However, combining [6,7] together with the analysis of the convergence of the exact CGL in [14] gives sufficient reasoning to relate A-norm of the error in the k-th iteration of finite-precision CGL computation with (an earlier) l-th iteration of exact computation with the same data as

$$\|\bar{x}_k^{\mathrm{L}} - x\|_A \approx \|x_l^{\mathrm{L}} - x\|_A .$$

The gap $k - l$ corresponds to the notion of the rank-deficiency of the computed matrix of Lanczos vectors or to the delay of convergence, see Fig. 1. Even though this idea has appeared in the literature repeatedly (see, e.g., [12, Sect. 5.9], [5, Chap. 3], [8, Sect. 6.7.4]) determination of the corresponding iterations $[k, l]$ is still an open question and can be highly problem-dependent. Furthermore, to the best of our knowledge, the possibility of comparison of other entities, especially those whose size does not decay monotonically, has not been addressed in literature.

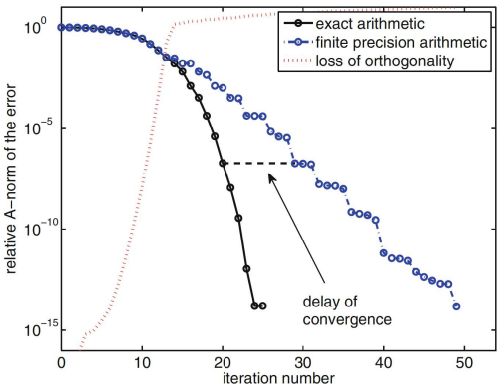


Fig. 1. Illustration of the loss of orthogonality and delay of convergence in CGL.

In this paper, we consider A symmetric positive-definite[1] and we exploit the idea of *simultaneous comparison* of finite-precision and exact computations for two related methods – CGL and MINRES. Since they form a pair of the norm-minimizing and the Galerkin method, we take advantage of their relationships proved in [2] to be approximately valid also in finite-precision computations. Because some finite-precision iterations k are a redundant consequence of reappearing information due to the delay of convergence, we do not consider all k. We rather assume we are given a subsequence $\{k_l\}_{l=1}^{m}$, $m \leq n$, where k_l is the finite-precision iteration related to the exact iteration l in the sense that the minimized quantities, i.e., A-norm of the error for CGL and residual norm for MINRES, are comparable between the two computations.[2] We show that some of the other entities cannot be compared directly. In particular, finite-precision CGL residuals (as well as their norms) and Lanczos vectors have to be *aggregated* over the intermediate iterations to form a counterpart to exact entities. We discuss influence of stagnation of MINRES on this comparison. Next, we discuss approaches to determine the subsequence $\{k_l\}$. Validity of obtained results is illustrated on numerical examples with matrices with various eigenvalue distribution.

The paper is organized as follows. Section 2 summarizes the Lanczos process and the two methods based on it - CGL and MINRES. Section 3 studies the relations between finite-precision and exact entities. Section 4 proposes some approaches to construct the subsequence $\{k_l\}$. Section 5 provides numerical experiments. Section 6 gives the conclusions. Throughout the paper, we assume $x_0 = 0$, i.e., $r_0 = b$; $\|\cdot\|, \|\cdot\|_A$ denotes the Euclidean and the energy norm respectively; e_j denotes the j-th column of the identity matrix of a suitable size. The entities computed in finite-precision arithmetic are denoted by bar.

[1] We only assume positive-definite matrices, so that the CGL iterations are well-defined in each step, although MINRES is well-defined also for indefinite matrices.

[2] The length of the subsequence, i.e., the index m, is typically determined by the iteration in which the finite-precision computation reaches the maximum attainable accuracy; see [12, Sect. 5.9.3].

2 Methods Based on Lanczos Tridiagonalization

Let $A \in \mathbb{R}^{n\times n}$ be a non-singular symmetric positive-definite matrix. Starting from a vector $v_1 = b/\delta_1$, $\delta_1 = \|b\|$, and initializing $v_0 = 0$, the tridiagonalization [11] computes, for $k = 1, 2, \ldots$,

$$\begin{aligned} \gamma_k &= (Av_k, v_k); \\ v_{k+1} &= Av_k - \gamma_k v_k - \delta_k v_{k-1}; \\ \delta_{k+1} &= \|v_{k+1}\|, \quad \text{if } \ \delta_{k+1} = 0, \ \text{then stop}; \\ v_{k+1} &= v_{k+1}/\delta_{k+1}. \end{aligned} \tag{2}$$

Vectors $v_1, \ldots, v_k$ form an orthonormal basis of the Krylov subspace (1). For simplicity of notation, we assume that the process (2) does not terminate before the iteration n, i.e., $\delta_{j+1} > 0$, $j = 1, \ldots n-1$. Denoting $V_k \equiv [v_1, \ldots, v_k] \in \mathbb{R}^{n\times k}$ and

$$T_k \equiv \begin{bmatrix} \gamma_1 & \delta_2 & & & \\ \delta_2 & \gamma_2 & \delta_3 & & \\ & \delta_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \delta_k \\ & & & \delta_k & \gamma_k \end{bmatrix} \in \mathbb{R}^{k\times k}, \qquad T_{k+1,k} \equiv \begin{bmatrix} T_k \\ e_k^T \delta_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k},$$

we can write the matrix formulation of the Lanczos tridiagonalization as

$$AV_k = V_k T_k + \delta_{k+1} v_{k+1} e_k^T = V_{k+1} T_{k+1,k} \quad k = 1, \ldots, n. \tag{3}$$

Based on [18], the Eq. (3) is in finite-precision replaced by

$$A\bar{V}_k = \bar{V}_k \bar{T}_k + \bar{\delta}_{k+1} \bar{v}_{k+1} e_k^T + \bar{F}_k = \bar{V}_{k+1} \bar{T}_{k+1,k} + \bar{F}_k, \quad k = 1, 2, \ldots,$$

where $\bar{F}_k$ is a small round-off term.

CGL and MINRES represent two methods based on the Lanczos tridiagonalization (2). At the k-th step, they search for the approximation of the solution in the subspace generated by the vectors $v_1, \ldots, v_k$, i.e., $x_k = V_k y_k$ for some $y_k \in \mathbb{R}^k$. The corresponding residual has the form

$$r_k \equiv b - Ax_k = b - AV_k y_k = V_{k+1}(\delta_1 e_1 - T_{k+1,k} y_k).$$

The CGL method as a Galerkin method imposes the orthogonality of the residuals yielding

$$T_k y_k^{\mathrm{L}} = \delta_1 e_1. \tag{4}$$

MINRES minimizes the norm of the residual r_k yielding

$$y_k^{\mathrm{M}} = \underset{y\in\mathbb{R}^k}{\operatorname{argmin}} \|\delta_1 e_1 - T_{k+1,k} y\|. \tag{5}$$

Table 1. An overview of the decay properties of various entities in CGL and MINRES computations with respect to (1). For more details see [4].

Method/quantity	$\|x_k - x\|$	$\|x_k - x\|_A$	$\|r_k\|$	$\|r_k\|_A$
CGL	Monotone	Minimized	–	–
MINRES	Monotone	Monotone	Minimized	–

Decay properties of various entities in CGL and MINRES are summarized in Table 1. The residual vectors of the two methods can be related as

$$r_k^{\mathrm{M}} = c_k^2 r_k^{\mathrm{L}} + s_k^2 r_{k-1}^{\mathrm{M}} \quad \text{with} \quad s_k^2 = \frac{\|r_k^{\mathrm{M}}\|^2}{\|r_{k-1}^{\mathrm{M}}\|^2}, \tag{6}$$

where s_k and c_k are the sine and cosine of the last Givens rotation used to eliminate the subdiagonal entry of the tridiagonal matrix $T_{k+1,k}$. The residual norms are related by the so-called *peak-plateau relation*

$$\|r_k^{\mathrm{L}}\| = \frac{\|r_k^{\mathrm{M}}\|}{\sqrt{1 - (\|r_k^{\mathrm{M}}\|/\|r_{k-1}^{\mathrm{M}}\|)^2}}, \tag{7}$$

or recursively, for $p = 0, 1, \ldots,$

$$\frac{1}{\sqrt{\sum_{j=k}^{k+p} 1/\|r_j^{\mathrm{L}}\|^2}} = \frac{\|r_{k+p}^{\mathrm{M}}\|}{\sqrt{1 - (\|r_{k+p}^{\mathrm{M}}\|/\|r_{k-1}^{\mathrm{M}}\|)^2}}. \tag{8}$$

For more details see [2].

3 Comparison of Finite-Precision and Exact Entities

In this section, we show how the vectors r_l^{L} and v_l can be compared to their finite-precision counterparts based on the relations between CGL and MINRES residuals. We assume that we have a sequence $\{k_l\}_{l=1}^m$ satisfying

$$\|x_l^{\mathrm{L}} - x\|_A \approx \|\bar{x}_{k_l}^{\mathrm{L}} - x\|_A, \qquad \|r_l^{\mathrm{M}}\| \approx \|\bar{r}_{k_l}^{\mathrm{M}}\|, \tag{9}$$

$$x_l^{\mathrm{L}} \approx \bar{x}_{k_l}^{\mathrm{L}}, \qquad r_l^{\mathrm{M}} \approx \bar{r}_{k_l}^{\mathrm{M}}, \tag{10}$$

i.e., the quantities minimized in exact arithmetic and the corresponding vectors are comparable to their finite-precision counterparts, in the sense that the distance between them measured in a suitable norm is small relative to their size. The first of the assumptions is based on the observation made in [12, Sect. 5.9], where the question of the delay of CG convergence and the associated rank deficiency of the computed subspace has been addressed. Approaches to find such a sequence are discussed in Sect. 4. We consider problems where T_k is not too badly conditioned and thus the error due to the inexact solution of (4) and (5) is negligible, and

$$r_k^{\mathrm{M,L}} = \bar{V}_{k+1}(\|b\|e_1 - \bar{T}_{k+1,k} y_k^{\mathrm{M,L}}) - \bar{F}_k y_k^{\mathrm{M,L}}. \tag{11}$$

Sects. 3.1–3.2 consider the case when the exact MINRES does not stagnate, i.e., $\|r_l^{\mathrm{M}}\|/\|r_{l-1}^{\mathrm{M}}\| \not\approx 1$ and subsequently $\|\bar{r}_{k_l}^{\mathrm{M}}\|/\|\bar{r}_{k_{l-1}}^{\mathrm{M}}\| \not\approx 1$. The other case is discussed in Sect. 3.3.

3.1 CGL Residual Norms and Vectors

First, we relate the CGL *residual norms*. Provided that MINRES does not stagnate, we obtain from (9) that

$$\frac{\|r_l^{\mathrm{M}}\|}{\sqrt{1-(\|r_l^{\mathrm{M}}\|/\|r_{l-1}^{\mathrm{M}}\|)^2}} \approx \frac{\|\bar{r}_{k_l}^{\mathrm{M}}\|}{\sqrt{1-(\|\bar{r}_{k_l}^{\mathrm{M}}\|/\|\bar{r}_{k_{l-1}}^{\mathrm{M}}\|)^2}}, \tag{12}$$

where the error of approximation is determined by the error of approximation in (9). Since $\|\bar{r}_{k_l}^{\mathrm{M}}\|/\|\bar{r}_{k_{l-1}}^{\mathrm{M}}\| \not\approx 1$, applying the technique from [2, Theorem 4], we see that (8) is approximately valid also in finite-precision computation, i.e.,

$$\frac{1}{\sqrt{\sum_{j=k_{l-1}+1}^{k_l} 1/\|\bar{r}_j^{\mathrm{L}}\|^2}} \approx \frac{\|\bar{r}_{k_l}^{\mathrm{M}}\|}{\sqrt{1-(\|\bar{r}_{k_l}^{\mathrm{M}}\|/\|\bar{r}_{k_{l-1}}^{\mathrm{M}}\|)^2}}, \tag{13}$$

where the error of approximation is determined by the round-off terms established in [2] not related to (9). Combining (12) and (13) with (7), we conclude that

$$\|r_l^{\mathrm{L}}\| \approx \frac{1}{\sqrt{\sum_{j=k_{l-1}+1}^{k_l} 1/\|\bar{r}_j^{\mathrm{L}}\|^2}}. \tag{14}$$

In words, the CGL residual norms cannot be compared directly, but finite-precision norms have to be *aggregated over the intermediate iterations*.

Now we turn to CGL *residual vectors*. Combining (6) and (7), we obtain

$$\frac{1}{\|r_l^{\mathrm{L}}\|^2}\, r_l^{\mathrm{L}} = \frac{1}{\|r_l^{\mathrm{M}}\|^2}\, r_l^{\mathrm{M}} - \frac{1}{\|r_{l-1}^{\mathrm{M}}\|^2}\, r_{l-1}^{\mathrm{M}} \tag{15}$$

for the exact arithmetic. Since we assume that (4) and (5) are solved with a negligible error, the first relation in (6) becomes in finite-precision computation

$$\bar{c}_k^2 \bar{r}_k^{\mathrm{L}} = \bar{r}_k^{\mathrm{M}} - \bar{s}_k^2 \bar{r}_{k-1}^{\mathrm{M}} + \bar{F}_k \bar{y}_k^{\mathrm{M}}. \tag{16}$$

Using [6], $\bar{s}_k$ and $\bar{c}_k$ can be expressed via the residual norms obtained from the exact computation with the extended matrix $\hat{A}$ in the same way as in the second equation of (6). Due to [2, Lemmas 4 and 8], these norms are approximately equal to those obtained by finite-precision computation. Using the relation (16) recursively, applying the results by Cullum and Greenbaum, and omitting the round-off terms, we obtain

$$\sum_{j=k_{l-1}+1}^{k_l} \frac{1}{\|\bar{r}_j^{\mathrm{L}}\|^2}\, \bar{r}_j^{\mathrm{L}} \approx \frac{1}{\|\bar{r}_{k_l}^{\mathrm{M}}\|^2}\, \bar{r}_{k_l}^{\mathrm{M}} - \frac{1}{\|\bar{r}_{k_{l-1}}^{\mathrm{M}}\|^2}\, \bar{r}_{k_{l-1}}^{\mathrm{M}}, \tag{17}$$

with the error of approximation determined by the round-off terms in [2]. Combining (15) and (17) while taking into account assumptions (9) and (10) on MINRES gives

$$\frac{1}{\|r_l^{\mathrm{L}}\|^2}\, r_l^{\mathrm{L}} \approx \sum_{j=k_{l-1}+1}^{k_l} \frac{1}{\|\bar{r}_j^{\mathrm{L}}\|^2}\, \bar{r}_j^{\mathrm{L}}.$$

Using the relation (14) we finally get

$$r_l^{\mathrm{L}} \approx \frac{1}{\sum_{j=k_{l-1}+1}^{k_l} \frac{1}{\|\bar{r}_j^{\mathrm{L}}\|^2}} \cdot \sum_{j=k_{l-1}+1}^{k_l} \frac{1}{\|\bar{r}_j^{\mathrm{L}}\|^2}\, \bar{r}_j^{\mathrm{L}}. \tag{18}$$

Thus the residual vectors from finite-precision computation have to be aggregated over the same iterations as their norms.

3.2 Lanczos Vectors

Recall that in exact arithmetic, the residual of CGL is a multiple of the subsequent Lanczos vector, i.e., $r_l^{\mathrm{L}} = (-1)^l \|r_l^{\mathrm{L}}\| v_{l+1}$. In finite-precision computation, (11) gives

$$\bar{r}_k^{\mathrm{L}} \approx (-1)^k \|\bar{r}_k^{\mathrm{L}}\| \bar{v}_{k+1}.$$

This, together with (14) and (18) yields

$$v_{l+1} \approx \frac{(-1)^l}{\sqrt{\sum_{j=k_{l-1}+1}^{k_l} \frac{1}{\|\bar{r}_j^{\mathrm{L}}\|^2}}} \cdot \sum_{j=k_{l-1}+1}^{k_l} \frac{(-1)^j}{\|\bar{r}_j^{\mathrm{L}}\|}\, \bar{v}_{j+1}. \tag{19}$$

Thus if the exact MINRES does not stagnate, assuming (9) and (10) the exact Lanczos vectors can be *approximated by a linear combination* of several consecutive Lanczos vectors from finite-precision computation. The derivation above does not rely on the orthogonality among the vectors $\bar{v}_{k_{l-1}+2}, \ldots, \bar{v}_{k_l+1}$.

3.3 Influence of Exact MINRES Stagnation

Now consider a plateau in exact MINRES convergence curve, i.e., $\|r_l^{\mathrm{M}}\|/\|r_{l-1}^{\mathrm{M}}\| \approx 1$ for some l. This can be caused (among others) by presence of a *tight cluster of eigenvalues* in the spectrum of A. Due to (7), we simultaneously observe a peak in the exact CGL residual norms. In this case, (12) and (13) may not hold and some of the CGL residual norms in exact arithmetic may not have a finite-precision counterpart of the form (14).

If the exact MINRES does not stagnate till the last iteration, we can proceed p_l iterations forward to achieve

$$\|r_{l+p_l}^{\mathrm{M}}\|/\|r_{l-1}^{\mathrm{M}}\| \ll 1. \tag{20}$$

Then, the approximations (12) and (13) become valid again for $\bar{r}^{\mathrm{M}}_{k_{l-1}}$ and $\bar{r}^{\mathrm{M}}_{k_{l+p_l}}$. Using (8), we conclude that the norms can be compared as

$$\frac{1}{\sqrt{\sum_{j=l}^{l+p_l} 1/\|r^{\mathrm{L}}_j\|^2}} = \frac{\|r^{\mathrm{M}}_{l+p_l}\|}{\sqrt{1-(\|r^{\mathrm{M}}_{l+p_l}\|/\|r^{\mathrm{M}}_{l-1}\|)^2}} \approx \frac{\|\bar{r}^{\mathrm{M}}_{k_{l+p_l}}\|}{\sqrt{1-(\|\bar{r}^{\mathrm{M}}_{k_{l+p_l}}\|/\|\bar{r}^{\mathrm{M}}_{k_{l-1}}\|)^2}} \approx \frac{1}{\sqrt{\sum_{j=k_{l-1}+1}^{k_{l+p_l}} 1/\|\bar{r}^{\mathrm{L}}_j\|^2}}, \quad (21)$$

i.e., both finite-precision and exact-arithmetic norms are aggregated over consecutive iterations. Other ways of comparison are also possible.

4 Construction of the Subsequence $\{k_l\}_{l=1}^m$

In this section, we aim at finding the subsequence of iterations $\{k_l\}_{l=1}^m$. We discuss several possible approaches, where the first one was in a similar form considered previously in [5,12].

Numerical Rank of the Computed Subspace. Focusing on the rank-deficiency of the matrix $\bar{V}_k$ of computed Lanczos vectors, the subsequence can be determined as

$$k_l^{\mathrm{rank}} \equiv \max\{k|\mathrm{num\_rank}(\bar{V}_k) = l\}.$$

The definition of numerical rank is generally a subtle issue and the resulting subsequence is dependent on its choice. Denoting $\bar{\sigma}_i$ the singular values of $\bar{V}_k$, we use

$$\mathrm{num\_rank}(\bar{V}_k) \equiv \{\#\bar{\sigma}_i|\, \bar{\sigma}_i > \tau\}. \quad (22)$$

The choice of the truncation parameter τ should reflect the fact that the exact matrix V_j has orthonormal columns, i.e., its singular values equal 1. We set in our experiments $\tau = 0.1$. Alternatively, numerical rank could be based, e.g., on finding the maximum gap between the singular values of $\bar{V}_k$.

Explicit Fitting of the Convergence Curves. Focusing on the delay of convergence, the subsequence can be found by explicit fitting of the quantities minimized over the Krylov subspace. In this way we find optimal subsequence with respect to one of the two assumptions in (9).

Fitting the CGL Convergence Curves:

$$k_l^{\mathrm{L}} = \operatorname*{argmin}_k \left| \|x^{\mathrm{L}}_l - x\|_A - \|\bar{x}^{\mathrm{L}}_k - x\|_A \right|. \quad (23)$$

Fitting the MINRES Convergence Curves:

$$k_l^{\mathrm{M}} = \operatorname*{argmin}_k |\|r^{\mathrm{M}}_l\| - \|\bar{r}^{\mathrm{M}}_k\||. \quad (24)$$

Note that the applicability of the approaches depends on the entities available. While (22) requires only the matrix $\bar{V}_k$, (24) requires r_l^{M}, i.e., exact computation has to be simulated. Approach (23) requires in addition the true solution x, which is not available in practical computations. Since CGL and MINRES are closely related, it is natural to expect that (23) and (24) provide similar subsequences. Fitting other entities, such as Ritz values, would theoretically also be possible. Note that (22) gives a strictly increasing subsequence $\{k_l\}_{l=1}^{m}$, which is not necessary the case for the other approaches and which becomes important especially for problems with stagnation in the exact convergence curves; see Sect. 5.

5 Numerical Experiments

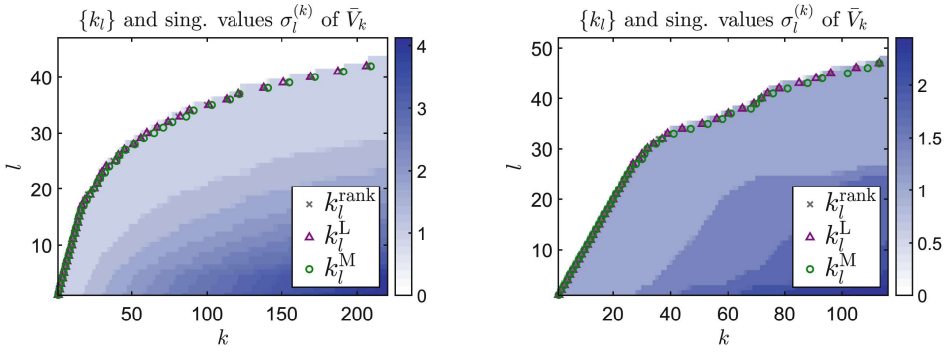
Now we compare the approaches to construction of $\{k_l\}$, we discuss the assumptions (9) and (10), and illustrate the results obtained for the CGL residuals and Lanczos vectors. Exact arithmetic is simulated by incorporating double reorthogonalization of the computed Lanczos vectors into the Lanczos process. It was shown in [18] that such algorithm is backward stable, i.e., it represents an exact Lanczos process for a nearby problem. The projected problems (4) and (5) are solved by the MATLAB function `mldivide`. Computations are stopped before the maximum attainable accuracy is reached. Experiments are performed in MATLAB R2015b.

We consider several test matrices from the Harwell-Boeing Collection [3] and the test matrix `strakos` introduced in [23], with parameters $n = 100$, $\lambda_{\min} = 0.1$, $\lambda_{\max} = 1000$, and $\gamma = 0.7$. The properties of the matrices are summarized in Table 2. For all matrices we choose $b = [1, \ldots, 1]^T$.

Fulfilling the Assumptions. In order to apply the results of Sect. 3, we first need the subsequence $\{k_l\}$ fulfilling (9) and (10). Figure 2 shows the subsequences constructed by approaches proposed in Sect. 4 together with the evolution of the singular values of the matrix V_k for problems `strakos` and `bcsstk01`. All three subsequences follow the edge of nonzero singular values throughout the whole computation. From the differences between $\{k_l^{\mathrm{L}}\}$ and $\{k_l^{\mathrm{M}}\}$ optimal with respect to the two convergence curves, we conclude that there is no $\{k_l\}$ optimal in all considered aspects. The plots in Fig. 3 (left) show the match between the exact

Table 2. Properties of the test matrices.

Problem	n	$\mathrm{nnz}(A)$	$\|A\|$	$\kappa(A)$
`strakos`	100	100	1×10^4	1×10^5
`bcsstk01`	48	400	3×10^9	1.6×10^6
`bcsstk04`	132	3648	9.6×10^6	5.6×10^6
`nos7`	729	4617	9.9×10^6	4.1×10^9



(a) `strakos` (b) `bcsstk01`

Fig. 2. The singular values of the computed matrix $\bar{V}_k$ together with the subsequences $\{k_l\}$ constructed using the three approaches from Sect. 4.

and finite-precision convergence curves shifted using $\{k_l\}$. We see a nice overlap of the convergence curves. In the right plots, we observe that

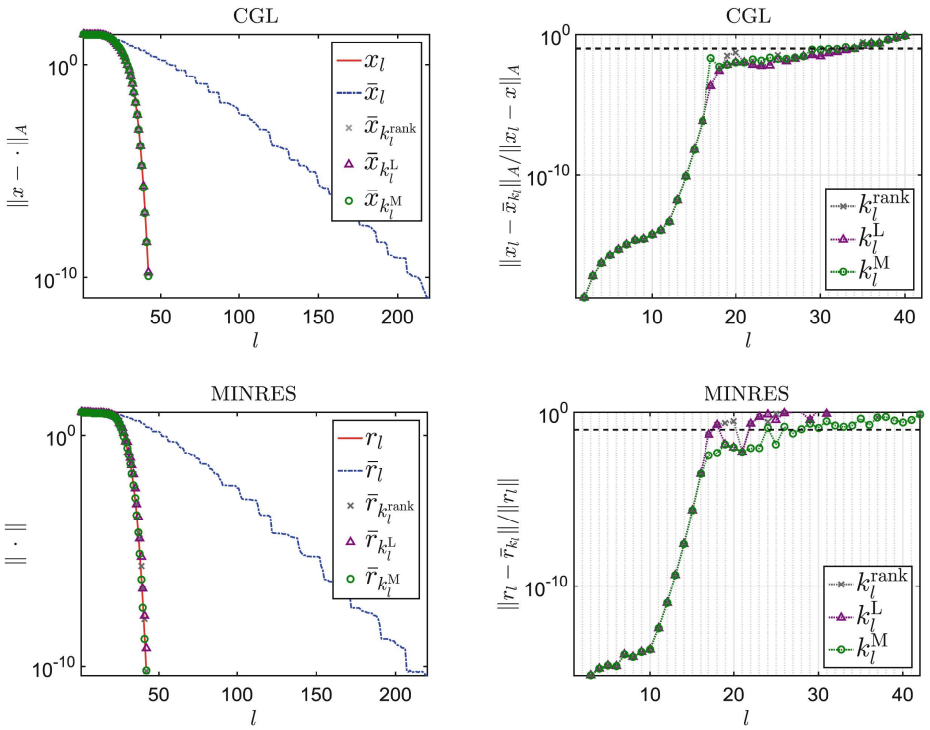
$$\|x_l^{\text{L}} - \bar{x}_{k_l}^{\text{L}}\|_A \ll \|x_l^{\text{L}} - x\|_A \quad \text{and} \quad \|r_l^{\text{M}} - \bar{r}_{k_l}^{\text{M}}\| \ll \|r_l^{\text{M}}\|$$

holds for most iterations, fulfilling sufficiently the assumption (10). From the experiments, the subsequence $\{k_l^{\text{M}}\}$ obtained by optimal fitting of the MINRES residual norms seems to provide the best results with respect to (9) and (10) and therefore it is used in the following experiments.

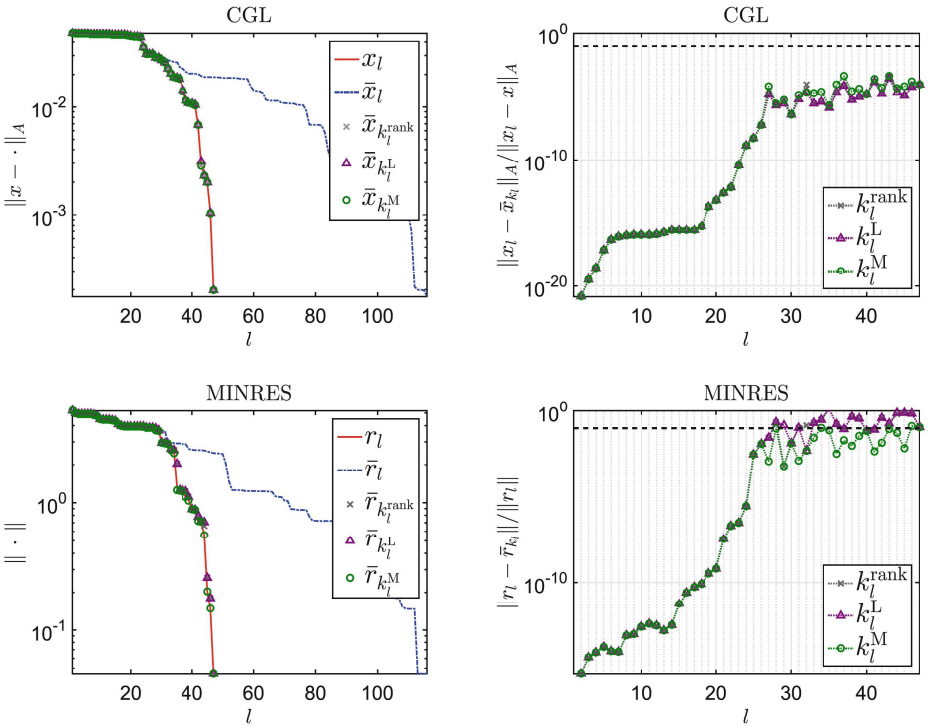
CGL Residuals and Lanczos Vectors. Now we verify (14), (18), and (19) derived in Sect. 3. Figure 4 (left) gives the comparison between the exact and finite-precision residual norms aggregated using (14) and shifted as in Fig. 3. In both cases, we observe very good match between the exact and aggregated finite-precision CGL convergence curve. For problem `strakos`, the approximation error of (18) for the residual vectors depicted in Fig. 4a (right) is essentially determined by the approximation error of the MINRES residual vectors, shown in Fig. 3a (right). For `bcsstk01`, the approximation is for the CGL residuals slightly worse than for the MINRES residuals, compare Figs. 3b and 4b. This is caused by the fact that in several iterations MINRES almost stagnates. A similar plot for the Lanczos vectors using (19) is provided in Fig. 5. Due to the relation (11), the approximation error is here similar to the approximation error of the CGL residuals.

Stagnation in MINRES Convergence. For real problems, the exact MINRES residual norm may not decrease sufficiently in each iteration, and severe oscillation may appear in the norm of the exact CGL residual.[3] Figure 6 shows results for two test problems of such type. Although (9) is satisfied, (14) does

[3] In such a case, approach (24) tends to construct subsequences for which $k_l = k_{l-1}$ may hold for some l.

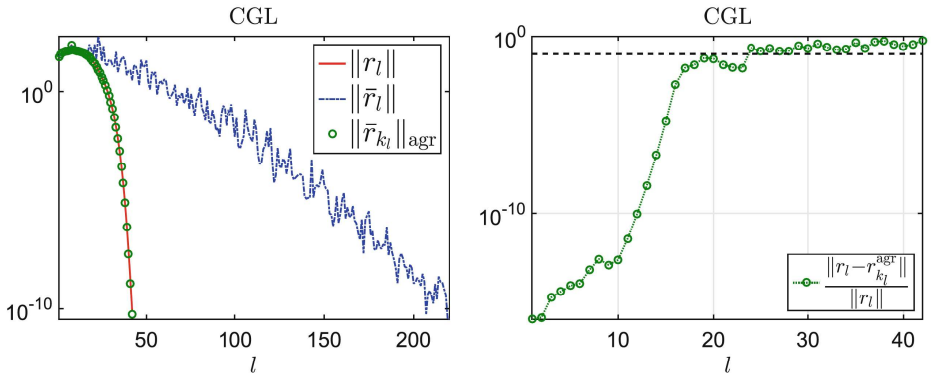


(a) `strakos`

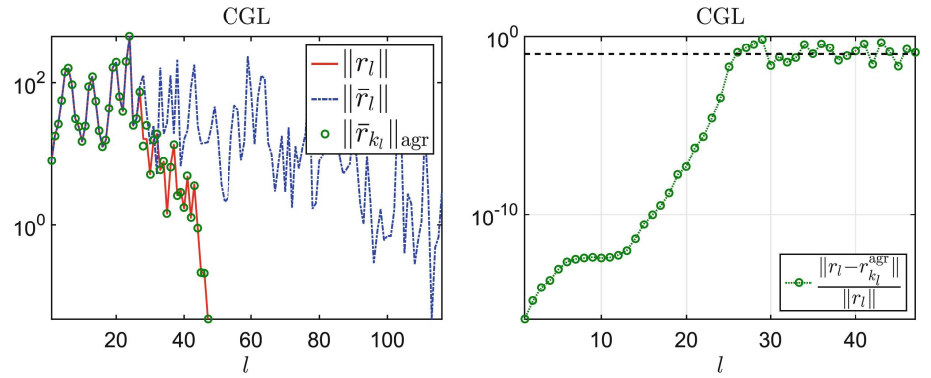


(b) `bcsstk01`

Fig. 3. Fulfillment of the assumptions (9) and (10) for two test problems. Left: The match between the exact convergence curve and the finite-precision convergence curve shifted using various subsequences $\{k_l\}$. Right: The match between the vectors themselves.

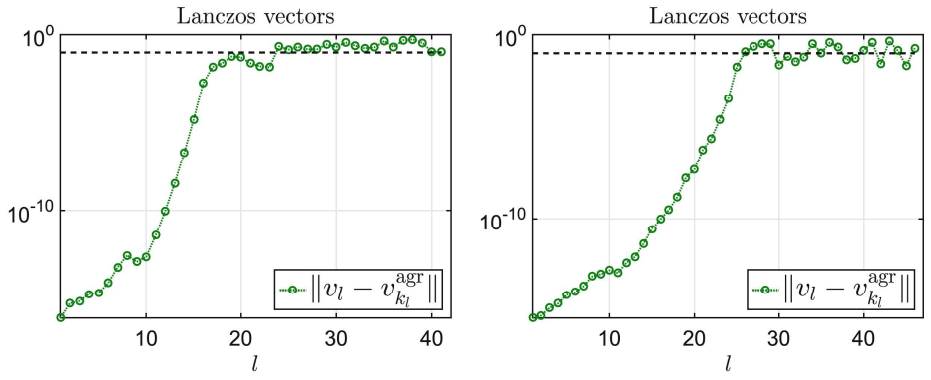


(a) `strakos`



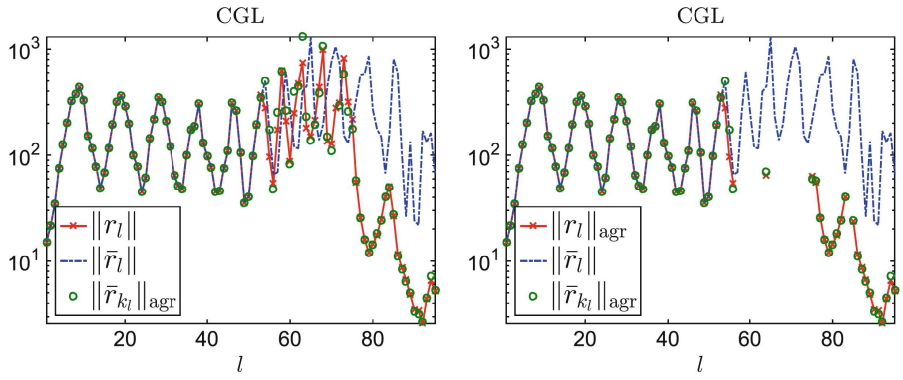
(b) `bcsstk01`

Fig. 4. Left: the exact and finite-precision CGL residual norms; $\|r_{k_l}\|_{\mathrm{agr}}$ denotes the right-hand side of (14). Right: relative difference between the exact and aggregated finite-precision residuals; $r_{k_l}^{\mathrm{agr}}$ denotes the right-hand side of (18).

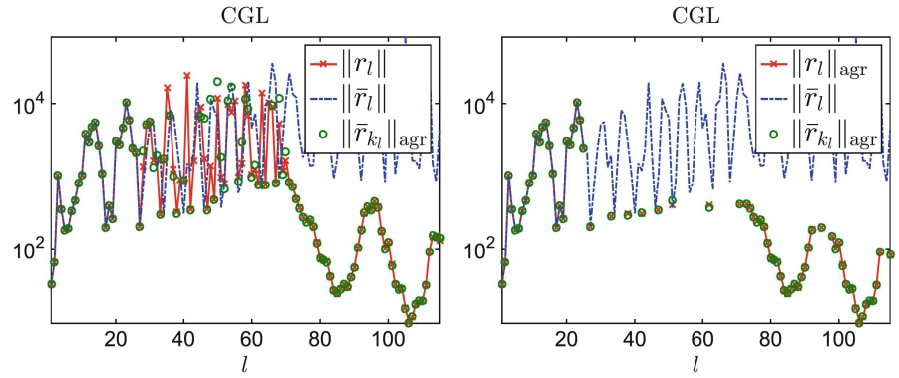


(a) `strakos` (b) `bcsstk01`

Fig. 5. Difference between the exact and aggregated finite-precision Lanczos vectors; $v_{k_l}^{\mathrm{agr}}$ denotes the right-hand side of (19).



(a) `bcsstk04`



(b) `nos7`

Fig. 6. CGL residuals compared as in (14) and (21) (plotted against $l+p_l$), respectively. Stagnation tolerance η is set to 0.01 (`bcsstk04`) and 0.002 (`nos7`).

not hold, see Fig. 6 (left). In order to apply the alternative formula derived in Sect. 3.3, we need to detect iterations with stagnation. Based on [2, Theorem 4], we suggest the criterion

$$1 - (\|r_l^{\mathrm{M}}\|/\|r_{l-1}^{\mathrm{M}}\|)^2 < \eta, \tag{25}$$

with η much smaller than the relative approximation error $\left|\|r_l^{\mathrm{M}}\| - \|\bar{r}_{k_l}^{\mathrm{M}}\|\right| / \|r_l^{\mathrm{M}}\|$. We proceed as follows. If the stagnation criterion (25) is satisfied, we substitute r_l^{M} by r_{l+1}^{M} and continue until we get sufficient decrease (20). Both sides of (21) are then plotted against $l + p_l$. We use every residual norm only once, i.e., in the next step of comparison we start with $l_{new} \leftarrow l + p_l + 1$. The larger the value η, the more iterations are aggregated and the sparser plots we get. Results obtained using this aggregation scheme are shown in Fig. 6 (right).

6 Conclusion

We have demonstrated that in many cases quantities minimized in exact MINRES and CGL computation can be compared directly to their selected

finite-precision counterparts for the same linear algebraic problem. We have proposed three approaches for determination of the subsequence of relevant finite-precision iterations – based on the numerical rank of the computed Lanczos matrix or on optimal fitting of the convergence curves. We have shown that entities whose size does not decay monotonically can not be compared directly. However, we have derived formulas relating the exact CGL residuals (and their norms) and the exact Lanczos vectors to vectors obtained in the finite-precision aggregated over the intermediate iterations. We have explained limitations of this approach for problems, where the exact MINRES method (nearly) stagnates and proposed an alternative way of comparison based on more general aggregation scheme. The results have been supported by experiments on standard test problems.

Acknowledgment. Research supported by the Grant Agency of Charles University (GAUK 196216) and by the Grant Agency of the Czech Republic (17-04150J).

References

1. Carpraux, J.F., Godunov, S.K., Kuznetsov, S.V.: Condition number of the Krylov bases and subspaces. Linear Algebra Appl. **248**, 137–160 (1996)
2. Cullum, J., Greenbaum, A.: Relations between Galerkin and norm-minimizing iterative methods for solving linear systems. SIAM J. Matrix Anal. Appl. **17**(2), 223–247 (1996)
3. Duff, I.S., Grimes, R.G., Lewis, J.G.: Users' guide for the Harwell-Boeing sparse matrix collection (1992)
4. Fong, D.C.L., Saunders, M.A.: CG versus MINRES: an empirical comparison. SQU J. Sci. **17**(1), 44–62 (2012)
5. Gergelits, T.: Analysis of Krylov subspace methods. Master's thesis, Charles University in Prague (2013)
6. Greenbaum, A.: Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences. Linear Algebra Appl. **113**, 7–63 (1989)
7. Greenbaum, A., Strakoš, Z.: Predicting the behavior of finite precision Lanczos and conjugate gradient computations. SIAM J. Matrix Anal. Appl. **13**(1), 121–137 (1992)
8. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems. Society for Industrial and Applied Mathematics (1998)
9. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Stand. **49**, 409–436 (1952)
10. Kuznetsov, S.V.: Perturbation bounds of the Krylov bases and associated Hessenberg forms. Linear Algebra Appl. **265**, 1–28 (1997)
11. Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J. Res. Natl. Bureau Stand. **45**, 255–282 (1950)
12. Liesen, J., Strakoš, Z.: Krylov Subspace Methods: Principles and Analysis. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2013)
13. Meurant, G., Strakoš, Z.: The Lanczos and conjugate gradient algorithms in finite precision arithmetic. Acta Numerica **15**, 471–542 (2006)

14. O'Leary, D.P., Strakoš, Z., Tichý, P.: On sensitivity of Gauss-Christoffel quadrature. Numer. Math. **107**(1), 147–174 (2007)
15. Paige, C.C.: The computation of eigenvalues and eigenvectors of very large sparse matrices. Ph.D. thesis, London University (1971)
16. Paige, C.C.: Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. Linear Algebra Appl. **34**, 235–258 (1980)
17. Paige, C.C., Saunders, M.A.: Solutions of sparse indefinite systems of linear equations. SIAM J. Numer. Anal. **12**(4), 617–629 (1975)
18. Paige, C.C.: Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. IMA J. Appl. Math. **18**(3), 341–349 (1976)
19. Paige, C.C.: An augmented stability result for the Lanczos Hermitian matrix tridiagonalization process. SIAM J. Matrix Anal. Appl. **31**(5), 2347–2359 (2010)
20. Paige, C.C., Van Dooren, P.: Sensitivity analysis of the Lanczos reduction. Numer. Linear Algebra Appl. **6**(1), 29–50 (1999)
21. Parlett, B.N., Scott, D.S.: The Lanczos algorithm with selective orthogonalization. Math. Comput. **33**(145), 217–238 (1979)
22. Simon, H.D.: Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. Linear Algebra Appl. **61**, 101–131 (1984)
23. Strakoš, Z.: On the real convergence rate of the conjugate gradient method. Linear Algebra Appl. **154–156**, 535–549 (1991)

4.2 Structure of the loss of orthogonality

In this section, we continue with the analysis of the Lanczos process described in Section 4.1. We focus on the relationship between the exact and the finite-precision Ritz vectors. Since both sets of vectors represent well-structured 'basis' of the underlying Krylov subspaces, we believe that detail understanding of their mutual relationship is crucial. We adopt the notation introduced in the previous section. Further, let

$$T_k = S^{(k)}\Theta^{(k)}(S^{(k)})^T \tag{4.1}$$

be the spectral decomposition of the symmetric tridiagonal matrix T_k, with $\Theta^{(k)} = \mathrm{diag}(\theta_1^{(k)}, \ldots, \theta_k^{(k)})$. We denote by $s_j^{(k)}$ the j-th column of $S^{(k)}$, i.e., the j-th eigenvector of T_k, and $s_{ij}^{(k)}$ the (i,j)-th entry of the matrix $S^{(k)}$. The j-th Ritz vector at the iteration k is defined as

$$y_j^{(k)} \equiv V_k s_j^{(k)}.$$

The corresponding entities in finite-precision computation are denoted analogously with a bar. By (λ_j, u_j) we denote the j-th eigenpair of A.

4.2.1 Finite-precision Ritz vectors

First, we recall some results regarding the convergence of the Ritz values and the Ritz vectors, and the loss of orthogonality among the Lanczos vectors in finite-precision arithmetic. Paige [1980] proved that for any Ritz pair $(\bar\theta_i^{(k)}, \bar y_i^{(k)})$ computed at the k-th step of the finite-precision Lanczos method it holds that

$$\min_{1\le j\le n} |\lambda_j - \bar\theta_i^{(k)}| \le \max\Big\{2.5(\bar\delta_{k+1}|\bar s_{ki}^{(k)}| + \sqrt{k}\|A\|\epsilon_1), \\ [(k+1)^3 + \sqrt{3}n^2]\|A\|\epsilon_2\Big\}, \tag{4.2}$$

$$\|\bar y_i^{(k)} - (u_j, \bar y_i^{(k)})u_j\| \le \frac{\bar\delta_{k+1}|\bar s_{ki}^{(k)}| + \sqrt{k}\|A\|\epsilon_1}{\min_{r\ne l}|\lambda_r - \bar\theta_i^{(k)}|}, \tag{4.3}$$

where both ϵ_1 and ϵ_2 are a small multiple of the machine precision ϵ_{mach}, see also [Meurant and Strakoš, 2006, sec. 4.2]. The inequality (4.2) implies that the size of $\bar\delta_{k+1}\bar s_{ki}^{(k)}$ indicates the convergence of $\bar\theta_i^{(k)}$ to an eigenvalue of A, which we denote by λ_l.[1] When in addition λ_l is well separated from the rest, it also indicates convergence of the Ritz vector to the corresponding eigenvector of A. Paige [1971] also proved that there is a structure in the loss of orthogonality, mainly that

$$(\bar v_{k+1}, \bar y_i^{(k)}) = \frac{\epsilon_{ii}^{(k)}}{\bar\delta_{k+1}\bar s_{ki}^{(k)}}, \tag{4.4}$$

[1] While small $\bar\delta_{k+1}|\bar s_{ki}^{(k)}|$ always ensures convergence of $\bar\theta_i^{(k)}$ to an eigenvalue of A, Wülling [2005] showed that in special cases, the opposite may not be true, i.e., closeness of $\bar\theta_i^{(k)}$ to some eigenvalue of A does not imply that $\bar\delta_{k+1}|\bar s_{kj}^{(k)}|$ is small.

where $|\epsilon_{ii}^{(k)}| \leq k\epsilon_2\|A\|$, i.e., that the loss of orthogonality of the newly generated Lanczos vector $\bar{v}_{k+1}$ in the direction of the Ritz vectors $\bar{y}_i^{(k)}$ can only be significant if $\bar{\delta}_{k+1}|\bar{s}_{ki}^{(k)}|$ or $\|\bar{y}_i^{(k)}\|$ is small. Combining (4.4) with other results, we can conclude that orthogonality can be lost only in the directions of converged Ritz vectors; see [Meurant and Strakoš, 2006, sec. 4.2].

Further, Paige investigated the loss of orthogonality between the Ritz vectors themselves, with respect to the distance of the corresponding Ritz values. In particular, he showed that

$$\begin{aligned}(\bar{\theta}_i^{(k)} - \bar{\theta}_j^{(k)})(\bar{y}_j^{(k)}, \bar{y}_i^{(k)}) = {} & \bar{\delta}_{k+1}(\bar{v}_{k+1}, \bar{s}_{kj}^{(k)}\bar{y}_i^{(k)} - \bar{s}_{ki}^{(k)}\bar{y}_j^{(k)}) \\ & + (\bar{F}_k\bar{s}_j^{(k)}, \bar{y}_i^{(k)}) - (\bar{F}_k\bar{s}_i^{(k)}, \bar{y}_j^{(k)}),\end{aligned} \tag{4.5}$$

where $\bar{F}_k = A\bar{V}_k - (\bar{V}_k\bar{T}_k + \bar{\delta}_{k+1}\bar{v}_{k+1}e_k^T)$; see [Paige, 1971, p. 113]. Substituting (4.4) into (4.5), we obtain

$$\begin{aligned}(\bar{\theta}_i^{(k)} - \bar{\theta}_j^{(k)})(\bar{y}_j^{(k)}, \bar{y}_i^{(k)}) = {} & \epsilon_{ii}^{(k)}\bar{s}_{kj}^{(k)}/\bar{s}_{ki}^{(k)} - \epsilon_{jj}^{(k)}\bar{s}_{ki}^{(k)}/\bar{s}_{kj}^{(k)} \\ & + (\bar{F}_k\bar{s}_j^{(k)}, \bar{y}_i^{(k)}) - (\bar{F}_k\bar{s}_i^{(k)}, \bar{y}_j^{(k)});\end{aligned} \tag{4.6}$$

see [Paige, 1971, p. 114]. Among others, the relation (4.6) implies that if $\bar{\theta}_i^{(k)}$ and $\bar{\theta}_j^{(k)}$ have not converged (i.e., $\bar{s}_{ki}^{(k)}$ and $\bar{s}_{kj}^{(k)}$ are not very small), and if they are not too close to each other, then the corresponding Ritz vectors will be almost orthogonal; see also [Parlett and Scott, 1979, sec. 4].

Note that convergence of a Ritz value may not necessarily lead to the loss of orthogonality in the direction of the corresponding Ritz vector, which is a common misunderstanding. Despite the equality sign in (4.4), the actual size of $(\bar{v}_{k+1}, \bar{y}_j^{(k)})$ remains unknown, since we only have an *upper bound* for the size of the numerator $\epsilon_{ii}^{(k)}$. We demonstrate on the following example that in special cases, $\epsilon_{ii}^{(k)}$ may vanish for all k and i, meaning that no loss of orthogonality occurs throughout the computation. Let T_n be the ultimate Jacobi matrix generated by the exact Lanczos process applied to the matrix A and the starting vector b, where we assume that the process does not terminate before the dimension of the matrix A is reached. We generally have no control over the loss of orthogonality in the finite-precision Lanczos process applied to A and b. However, when we apply the Lanczos process to T_n and the starting vector e_1, in all reasonable computational environments no loss of orthogonality will occur, despite the fact that the eigenvalues of T_n are exactly those of A. This is caused by the special structure of the matrix T_n and the starting vector e_1, making the Lanczos process simply copy the entries of the input matrix to the intermediate tridiagonal matrices generated in the Lanczos process, effectively avoiding any rounding errors.

4.2.2 Relationship between the exact and finite-precision Ritz vectors

While Paige investigated the finite-precision Lanczos process on its own, we are more interested in its relationship to the exact Lanczos process. The following

proposition uses the structure of the proof of (4.5) to investigate the loss of orthogonality between the exact and finite-precision Ritz vectors.

Proposition 4.1. *Let*

$$AV_l = V_l T_l + \delta_{l+1} v_{l+1} e_l^T \tag{4.7}$$
$$A\bar{V}_k = \bar{V}_k \bar{T}_k + \bar{\delta}_{k+1} \bar{v}_{k+1} e_k^T + \bar{F}_k \tag{4.8}$$

represent the l-th and k-th step of the exact and the finite-precision Lanczos process, respectively. Using the notation introduced above, it holds that

$$\begin{aligned}(\bar{\theta}_i^{(k)} - \theta_j^{(l)})(y_j^{(l)}, \bar{y}_i^{(k)}) = \delta_{l+1} s_{lj}^{(l)} (\bar{y}_i^{(k)}, v_{l+1}) - \bar{\delta}_{k+1} \bar{s}_{ki}^{(k)} (y_j^{(l)}, \bar{v}_{k+1}) \\ - (\bar{F}_k \bar{s}_i^{(k)}, y_j^{(l)}).\end{aligned} \tag{4.9}$$

Proof. Multiplying (4.7) by $\bar{V}_k^T$ and (4.8) by V_l^T from the left, we obtain

$$\begin{aligned}\bar{V}_k^T A V_l &= \bar{V}_k^T V_l T_l + \delta_{l+1} \bar{V}_k^T v_{l+1} e_l^T, \\ V_l^T A \bar{V}_k &= V_l^T \bar{V}_k \bar{T}_k + \bar{\delta}_{k+1} V_l^T \bar{v}_{k+1} e_k^T + V_l^T \bar{F}_k.\end{aligned}$$

Since A is symmetric, $V_l^T A \bar{V}_k = (\bar{V}_k^T A V_l)^T$, and therefore

$$V_l^T \bar{V}_k \bar{T}_k + \bar{\delta}_{k+1} V_l^T \bar{v}_{k+1} e_k^T + V_l^T \bar{F}_k = T_l V_l^T \bar{V}_k + \delta_{l+1} e_l v_{l+1}^T \bar{V}_k.$$

Using the spectral decomposition of the Jacobi matrices T_l and $\bar{T}_k$ (omitting the superscripts in the decomposition (4.1)) yields

$$V_l^T \bar{V}_k \bar{S} \bar{\Theta} \bar{S}^T + \bar{\delta}_{k+1} V_l^T \bar{v}_{k+1} e_k^T + V_l^T \bar{F}_k = S \Theta S^T V_l^T \bar{V}_k + \delta_{l+1} e_l v_{l+1}^T \bar{V}_k.$$

Multiplying the equation by S^T from the left and by $\bar{S}$ from the right, we have

$$S^T V_l^T \bar{V}_k \bar{S} \bar{\Theta} + \bar{\delta}_{k+1} S^T V_l^T \bar{v}_{k+1} e_k^T \bar{S} + S^T V_l^T \bar{F}_k \bar{S} = \Theta S^T V_l^T \bar{V}_k \bar{S} + \delta_{l+1} S^T e_l v_{l+1}^T \bar{V}_k \bar{S},$$

which, using $Y = V_l S$ and $\bar{Y} = \bar{V}_k \bar{S}$, and rearranging the equations gives

$$Y^T \bar{Y} \bar{\Theta} - \Theta Y^T \bar{Y} = \delta_{l+1} S^T e_l v_{l+1}^T \bar{Y} - \bar{\delta}_{k+1} Y^T \bar{v}_{k+1} e_k^T \bar{S} - Y^T \bar{F}_k \bar{S}.$$

Multiplying both sides by the corresponding canonical vectors as $e_j^T(\cdot)e_i$ yields (4.9). □

Proposition 4.1 connects the loss of orthogonality between the Ritz vectors with quantities which are interesting on their own:

- Distance of the computed and the exact Ritz values, $|\bar{\theta}_i^{(k)} - \theta_j^{(l)}|$.
- Scalars $\delta_{l+1} s_{lj}^{(l)}$ and $\bar{\delta}_{k+1} \bar{s}_{ki}^{(k)}$ whose size indicates convergence of the corresponding Ritz value in the exact and the finite-precision computation, respectively.

- Angles between the newly generated Lanczos vector and a given Ritz vector, $(\bar{y}_i^{(k)}, v_{l+1})$ and $(y_j^{(l)}, \bar{v}_{k+1})$, mixing the exact and finite-precision computations.

We would like to identify the cases for which $(y_j^{(l)}, \bar{y}_i^{(k)})$ is small, i.e., cases when the finite-precision Ritz vector $\bar{y}_i^{(k)}$ (if nonvanishing) is close to orthogonal to the exact Ritz vector $y_j^{(l)}$. Since $\|\bar{s}_i^{(k)}\| = 1$ and $\|y_j^{(l)}\| = 1$, we have $|(\bar{F}_k \bar{s}_i^{(k)}, y_j^{(l)})| \leq \|\bar{F}_k\|$. Therefore, when the size of the two terms on the right-hand side of (4.9) is small while the size of $\bar{\theta}_i^{(k)} - \theta_j^{(l)}$ is not, then the size of $(y_j^{(l)}, \bar{y}_i^{(k)})$ becomes necessarily small as well. We investigate three possible scenarios, when this happens:

- If $\delta_{l+1}|s_{lj}^{(l)}|$ and $\bar{\delta}_{k+1}|\bar{s}_{ki}^{(k)}|$ are both small whereas $|\bar{\theta}_i^{(k)} - \theta_j^{(l)}|$ is not, i.e., $\bar{\theta}_i^{(k)}$ and $\theta_j^{(l)}$ have converged to two well separated eigenvalues of A, then the two Ritz vectors are essentially orthogonal (if nonvanishing). This relation is valid independent of the relation between k and l and is consistent with (4.2) and (4.3), because clearly two eigenvectors corresponding to two distinct eigenvalues of A are mutually orthogonal.

- If $k = k_l$, i.e., the iteration k in finite-precision arithmetic corresponds to the iteration l in exact arithmetic in the sense of [Gergelits et al., 2018, sec. 4], then v_{l+1} is supposed to be close to orthogonal to the columns of $\bar{V}_k$. Therefore any converged Ritz vector $\bar{y}_i^{(k)}$ in the finite-precision computation with small $\bar{\delta}_{k+1}|\bar{s}_{ki}^{(k)}|$ will be orthogonal (if nonvanishing) to all Ritz vectors $y_j^{(l)}$ (even unconverged) corresponding to the Ritz values well separated from that of $\bar{y}_i^{(k)}$. This corresponds to our intuition that if the exact computation is ahead of the finite-precision one, then converged finite-precision Ritz vectors have their counterparts in exact arithmetic. Orthogonality of the exact Ritz vectors then gives the orthogonality between the exact and the converged finite-precision Ritz vectors (again except those corresponding to the same cluster of eigenvalues of A).

- Converse formulation, i.e., based on the orthogonality between $\bar{v}_{k+1}$ and the columns of V_l would also be possible, however $V_l^T \bar{v}_{k+1} \approx 0$ is somewhat difficult to predict, except for special cases such as the one discussed at the end of Section 4.2.1.

Proposition 4.1 in its present form gives a rather incomplete description of the relation between the finite-precision and the exact Ritz vectors, however relations of type (4.4) rely on the symmetry of the matrix $\bar{V}_k^T \bar{V}_k$ and therefore can be directly reformulated neither for $(\bar{y}_i^{(k)}, v_{l+1})$ nor for $(y_j^{(l)}, \bar{v}_{k+1})$. Subsequently, extending the result to the form of (4.6) is not possible. Better understanding of the finite-precision and the exact Ritz vectors, and also the Lanczos vectors will be subject of our future research.

Bibliography

T. Gergelits. Analysis of Krylov subspace methods. Master's thesis, Charles University in Prague, 2013.

T. Gergelits, I. Hnětynková, and M. Kubínová. Relating computed and exact entities in methods based on Lanczos tridiagonalization. In T. Kozubek, M. Čermák, P. Tichý, R. Blaheta, J. Šístek, D. Lukáš, and J. Jaroš, editors, *High Performance Computing in Science and Engineering*, pages 73–87, Cham, 2018. Springer International Publishing.

A. Greenbaum. Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences. *Linear Algebra Appl.*, 113:7–63, 1989.

C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

J. Liesen and Z. Strakoš. *Krylov subspace methods*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2013.

G. Meurant and Z. Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

C. C. Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices*. PhD thesis, London University, 1971.

C. C. Paige. Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. *Linear Algebra Appl.*, 34:235–258, 1980.

B. N. Parlett and D. S. Scott. The Lanczos algorithm with selective orthogonalization. *Math. Comput.*, 33(145):217–238, 1979.

W. Wülling. On stabilization and convergence of clustered Ritz values in the Lanczos method. *SIAM J. Matrix Anal. Appl.*, 27(3):891–908, 2005.

5. Robust regression for mixed Poisson–Gaussian model

This chapter deals with inverse problems, where noise in the data comes from various sources – more specifically, we assume image deblurring problems with a combination of shot noise and read-out noise. In addition, we assume that the data is further corrupted by an unknown type of corruptions, generally referred to as outliers. Problems with mixed noise as well as problems with outliers have been studied extensively in the literature. However, to our knowledge, very little has been done for problems with data containing both issues at the same time. In the article Kubínová and Nagy [in press] included in Section 5.1, we derive a model that can be used to numerically deal with such type of corruptions. The model leads to a constrained optimization problem, which can be efficiently solved using a modification of Newton's method. In Section 5.2, we briefly comment on the possibilities of relaxing Newton's method to a version of the Gauss–Newton method and investigate, which loss functions are admissible for this scheme.

5.1 Article published in Numerical Algorithms

This section contains the article Kubínová and Nagy [in press]. Reprinted by permission from Springer Nature: *Numerical Algorithms*, Kubínová, M. & Nagy, J. G.: Robust regression for mixed Poisson–Gaussian model, copyright (2018), advance online publication, 19/01/2018 (doi:10.1007/s11075-017-0463-1).

Numer Algor
https://doi.org/10.1007/s11075-017-0463-1

ORIGINAL PAPER

Robust regression for mixed Poisson–Gaussian model

Marie Kubínová[1] · **James G. Nagy**[2]

Received: 17 May 2017 / Accepted: 19 December 2017
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract This paper focuses on efficient computational approaches to compute approximate solutions of a linear inverse problem that is contaminated with mixed Poisson–Gaussian noise, and when there are additional outliers in the measured data. The Poisson–Gaussian noise leads to a weighted minimization problem, with solution-dependent weights. To address outliers, the standard least squares fit-to-data metric is replaced by the Talwar robust regression function. Convexity, regularization parameter selection schemes, and incorporation of non-negative constraints are investigated. A projected Newton algorithm is used to solve the resulting constrained optimization problem, and a preconditioner is proposed to accelerate conjugate gradient Hessian solves. Numerical experiments on problems from image deblurring illustrate the effectiveness of the methods.

Keywords Poisson–Gaussian model · Weighted least squares · Robust regression · Preconditioner · Image restoration

Mathematics Subject Classification 2010 65N20 · 49M15 · 62F35

Research of the first author supported in part by the grant SVV-2017-260455. Research of the second author supported in part by US National Science Foundation under grant no. DMS-1522760.

✉ Marie Kubínová
kubinova@karlin.mff.cuni.cz

James G. Nagy
nagy@mathcs.emory.edu

[1] Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

[2] Department of Mathematics and Computer Science, Emory University, Atlanta, GA, USA

Published online: 19 January 2018

1 Introduction

In this paper, we consider efficient computational approaches to compute approximate solutions of a linear inverse problem,

$$b = Ax_{\text{true}} + \eta, \qquad A \in \mathbb{R}^{m\times n}, \tag{1}$$

where A is a known matrix, vector b represents known acquired data, η represents noise, and vector x_{true} represents the unknown quantity that needs to be approximated. We are particularly interested in imaging applications where $x_{\text{true}} \geq 0$ and $Ax_{\text{true}} \geq 0$. Although this basic problem has been studied extensively (see, for example, [9, 16, 27, 34] and the references therein), the noise is typically assumed to come from a single source (or to be represented by a single statistical distribution) and the data to contain no outliers. In this paper, we focus on a practical situation that arises in many imaging applications, and for which relatively little work has been done, namely when the noise is comprised of a mixture of Poisson and Gaussian components *and* when there are outliers in the measured data. While some research has been done on the two topics separately (i.e., mixed Poisson–Gaussian noise models *or* outliers in measured data), to our knowledge, no work has been done when the measured data contains both issues. In the following, we review some of the approaches used to handle each of the issues.

1.1 Poisson–Gaussian noise

A Poisson–Gaussian statistical model for the measured data takes the form

$$b_i = n_{\text{obj}}(i) + g(i),\ i = 1,\dots,m, \qquad n_{\text{obj}}(i) \sim \text{Pois}([Ax_{\text{true}}]_i),\ g(i) \sim \mathcal{N}(0,\sigma^2), \tag{2}$$

where b_i is the ith component of the vector b and $[Ax_{\text{true}}]_i$ the ith component of the true noise-free data Ax_{true}. We assume that the two random variables $n_{\text{obj}}(i)$ and $g(i)$ are independent. This mixed noise model arises in many important applications, such as when using charged coupled device (CCD) arrays, x-ray detectors, and infrared photometers [2, 13, 22, 23, 32]. The Poisson part (sometimes referred to as shot noise) can arise from the accumulation of photons over a detector, and the Gaussian part usually is due to *read-out* noise from a detector, which can be generated by thermal fluctuations in interconnected electronics.

Since the log-likelihood function for the mixed Poisson–Gaussian model (2) has an infinite series representation [32], we assume a simplified model, where both random variables have the same type of distribution. There are two main approaches one can take to generate a simplified model. The first approach is to add σ^2 to each component of the vector b, and from (2), it then follows that

$$\mathbb{E}(b_i + \sigma^2) = [Ax_{\text{true}}]_i + \sigma^2 \quad \text{and} \quad \text{var}(b_i + \sigma^2) = [Ax_{\text{true}}]_i + \sigma^2.$$

For large σ, the Gaussian random variable $g(i) + \sigma^2$ is well-approximated by a Poisson random variable with the Poisson parameter σ^2, and therefore, $b_i + \sigma^2$ is also well approximated by a Poisson random variable with the Poisson parameter

$[Ax_{\text{true}}]_i + \sigma^2$. The data fidelity function corresponding to the negative Poisson log-likelihood then has the form

$$\sum_{i=1}^{m} ([Ax_{\text{true}}]_i + \sigma^2) - (b_i + \sigma^2) \log([Ax_{\text{true}}]_i + \sigma^2); \tag{3}$$

see also [32]. An alternative approach is to approximate the true negative log-likelihood by a weighted least-squares function, where the weights correspond to the measured data, i.e.,

$$\sum_{i=1}^{m} \frac{1}{2} \left(\frac{[Ax]_i - b_i}{\sqrt{b_i + \sigma^2}} \right)^2; \tag{4}$$

see [18, Sec. 1.3]. A more accurate approximation can be achieved by replacing the measured data by the computed data (which depends on x), i.e., replace the fidelity function (4) by

$$\sum_{i=1}^{m} \frac{1}{2} \left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}} \right)^2; \tag{5}$$

see [33] for more details. Additional additive Poisson noise (e.g., background emission) can be incorporated into the model in a straightforward way. We remark that there are also approaches to handle mixed Gaussian-impulsive noise; see for example [35, 36]. However, the focus in this paper is on the problem of mixed Gaussian–Poisson noise models.

1.2 Outliers

For data corrupted solely with Gaussian noise, i.e.,

$$b_i = [Ax]_i + g(i), \ i = 1, \ldots, m, \quad g(i) \sim \mathcal{N}(0, \sigma^2),$$

employing the negative log-likelihood leads to the standard least-squares functional

$$\sum_{i=1}^{m} \frac{1}{2} ([Ax]_i - b_i)^2 . \tag{6}$$

It is well known however that a computed solution based on least squares is not robust if outliers occur, meaning that even a small number of components with gross errors can cause a severe deterioration of our estimate. Robustness of the least squares fidelity function can be achieved by replacing the loss function $\frac{1}{2}t^2$ used in (6) by a function $\rho(t)$ as

$$\sum_{i=1}^{m} \rho ([Ax]_i - b_i) , \tag{7}$$

where the function ρ is less stringent towards the gross errors and satisfies the following conditions:

1. $\rho(t) \geq 0$;
2. $\rho(t) = 0 \Leftrightarrow t = 0$;
3. $\rho(-t) = \rho(t)$;
4. $\rho(t') \geq \rho(t)$, for $t' \geq t \geq 0$;

see also [18, Sec. 1.5]. A list of eight most commonly used loss functions ρ can be found in [6] or in MATLAB under `robustfit`; some of them are discussed in Section 2.1. Each of these functions also depends on a parameter β (see Section 2.2) defining the trade-off between the robustness and efficiency. Note that if we use this robust regression approach, in order to reduce the influence of possible outliers, we always sacrifice some efficiency at the model.

In this paper, we focus on combining these two approaches to suppress the influence of outliers for data with mixed noise (2). Our work has been motivated by O'Leary [30], and more recent work by Calef [5]. The initial ideas of our work were first outlined in the conference paper [21].

The paper is organized as follows. In Section 2, we introduce a data-fidelity function suitable for data corrupted both with mixed Poisson–Gaussian noise and outliers. In Section 3, we propose a regularization parameter choice method for the regularization of the resulting inverse problem, and in Section 4, we focus on the optimization algorithm and the solution of the linear subproblems. Section 5 demonstrates the performance of the resulting method on image deblurring problems with various types of outliers.

Throughout the paper, D (or D with an accent) denotes a general real diagonal matrix and e_i denotes the ith column of the identity matrix of a suitable size.

2 Data-fidelity function

In Section 1, we reviewed fidelity functions (3), (4), and (5), commonly used for problems with mixed Poisson–Gaussian noise and also robust loss functions used to handle problems with Gaussian noise and outliers (7). Since we need to deal with both issues simultaneously here, we propose combining both approaches. More specifically, combining a robust loss function with the weighted least squares problem (5), so that the data fidelity function becomes

$$J(x) = \sum_{i=1}^{m} \rho\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}} \right). \tag{8}$$

In the remainder of this section, we investigate the properties of the proposed data-fidelity function (8) and the choice of the robustness parameter β, which is defined in the next subsection.

2.1 Choice of the loss function—convexity analysis

For ordinary least squares, functions known under names Huber, logistic, Fair, and Talwar, shown in Fig. 1, lead to an interval-wise convex data fidelity function (see [30]), i.e., positive semi-definite Hessian, which is favorable for Newton-type minimization algorithms. This however does not always hold in our case where the weighted least squares formulation (8) has solution-dependent weights.

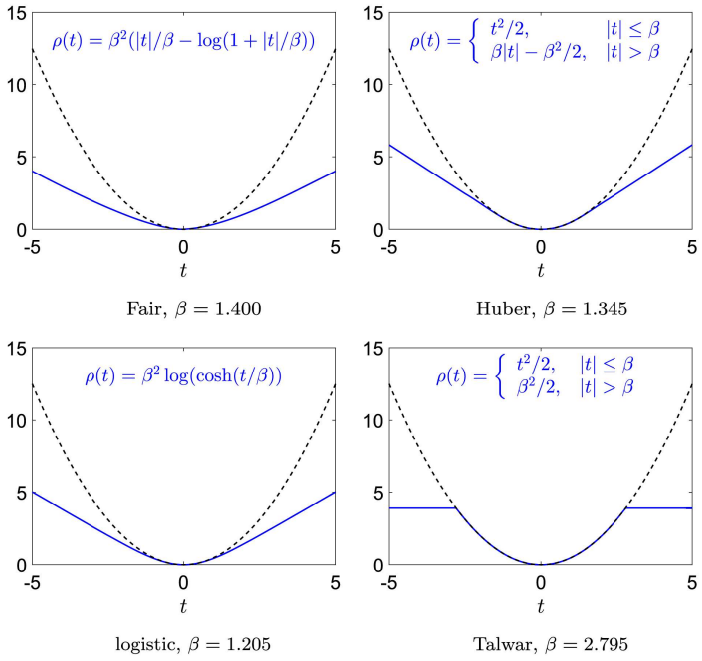


Fig. 1 Loss functions Fair, Huber, logistic, and Talwar with the tuning parameter β corresponding to 95% efficiency with respect to $\mathcal{N}(0, 1)$ (solid line), together with the standard loss function $t^2/2$ (dashed line)

From [33], we know that the functional (8) is convex for $\rho(t) = t^2/2$ and therefore has a positive definite Hessian. For a general function ρ, the gradient and the Hessian of the functional (8) become

$$\begin{aligned} \text{grad}_J(x) &= A^T z, \qquad z_i = \frac{\frac{1}{2}[Ax]_i + \frac{1}{2}b_i + \sigma^2}{([Ax]_i + \sigma^2)^{3/2}} \rho'\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right); \\ \text{Hess}_J(x) &= A^T D A, \quad D_{ii} = \frac{\left(\frac{1}{2}[Ax]_i + \frac{1}{2}b_i + \sigma^2\right)^2}{([Ax]_i + \sigma^2)^3} \rho''\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right) \\ &\quad - \frac{\left(\frac{1}{4}[Ax]_i + \frac{3}{4}b_i + \sigma^2\right)}{([Ax]_i + \sigma^2)^{5/2}} \rho'\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right); \end{aligned}$$

where for points in which the derivatives ρ' or ρ'' do not exist, these are substituted by left/right derivatives, similarly to [6]. We investigate the entries D_{ii} in order to examine the positive semi-definiteness of the Hessian $\text{Hess}_J(x)$. Recall that $A^T DA$ is positive semi-definite, if $D_{ii} \ge 0$. This condition is clearly satisfied for the loss function Talwar

$$\rho(t) = \begin{cases} t^2/2, & |t| \le \beta, \\ \beta^2/2, & |t| > \beta, \end{cases} \tag{9}$$

as the entries D_{ii} either coincide with those for $\rho(t) = t^2/2$ or are zero.

In the following, we show that no other function ρ from [6] can ensure non-negative D_{ii}. Since we assume $[Ax]_i, b_i \geq 0, \forall i$, it holds that $D_{ii} \geq 0$, if

$$\frac{\left([Ax]_i + 3b_i + 4\sigma^2\right)\left([Ax]_i + \sigma^2\right)^{1/2}}{\left([Ax]_i + b_i + 2\sigma^2\right)^2}\rho'\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right) \leq \rho''\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right). \tag{10}$$

We see immediately that ρ'' has to be non-negative. Moreover, (10) is difficult to achieve for $[Ax]_i \gg b_i + \sigma^2$, when

$$\frac{1}{\sqrt{[Ax]_i}}\rho'(\sqrt{[Ax]_i}) \lesssim \rho''(\sqrt{[Ax]_i}),$$

must hold. If ρ' is not zero, this corresponds to

$$\rho'(t) \gtrsim t \quad \text{yielding} \quad \rho(t) \gtrsim t^2/2,$$

i.e., for large $[Ax]_i$, the loss function ρ has to grow at least quadratically. Therefore, considering the functions from [6], the only loss function ρ for which the data fidelity function (8) has positive semidefinite Hessian is the function Talwar.

2.2 Selection of the robustness parameter

Introducing robustness with respect to outliers comes at a cost since the confidence interval for estimates involving robust loss functions ρ (7) is wider than for least squares (6), for normally distributed noise. This can be quantified by the so-called (relative) efficiency of a regression estimator, given as the ratio between its variance and that of the least squares estimator, for a sample with normally distributed errors. The efficiency is usually calculated in terms of asymptotic efficiency, i.e., with the size of the sample going to infinity. For more details see, e.g., [11, Section 4] and [31]. The asymptotic efficiency of the robust estimators is typically controlled by a parameter, here denoted by β. Parameters β providing 95% asymptotic efficiency for the standard robust loss functions can be found in [6]. For Talwar, the 95% efficiency tuning parameter is

$$\beta_{95} = 2.795. \tag{11}$$

Note that in our specific case, the random variable inside the function ρ in (8) is already rescaled to have unit variance and therefore approximately unit normal distribution. We may therefore apply the parameter β_{95} without any further rescaling based on estimated variance, which is usually required in case of ordinary least squares with unknown variance of noise. Function Talwar with $\beta = \beta_{95}$ is shown in the lower right panel of Fig. 1.

2.3 Non-negativity constraints

In many applications, such as imaging, the reconstruction will benefit from taking into account the prior information about the component-wise non-negativity of the

true solution x_{true}. Here, however, imposing non-negativity is not just a question of visual appeal, it also guarantees the two estimates (3) and (5) of the negative log-likelihood will provide similar results; see [33]. Therefore, the component-wise non-negativity constraint is an integral part of the resulting optimization problem. However, employment of the non-negativity constraint results in the need of more sophisticated optimization tools. The use of one of the possible algorithms is discussed in Section 4.

3 Regularization and selection of the regularization parameter

As a consequence of noise and ill-posedness of the inverse problem (1), some form of regularization needs to be employed in order to achieve a reasonable approximation of the true solution x_{true}. For computational convenience, we use Tikhonov regularization with a quadratic penalization term, i.e., we minimize the functional of the form

$$J_\lambda(x) \equiv \sum_{i=1}^{m} \rho\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right) + \frac{\lambda}{2}\|Lx\|^2, \qquad x \geq 0. \tag{12}$$

We assume that a good regularization parameter λ with respect to L is used so that the penalty term is reasonably close to the prior and the residual therefore is close to noise. In case of robust regression, it is particularly important not to over-regularize, since this would lead to large residuals and too many components of the data b would be considered outliers. Methods for choosing λ are discussed in this section.

3.1 Morozov's discrepancy principle

Since the residual components are scaled, and for data without outliers, we have the expected value

$$E\left\{\frac{1}{n}\sum_{i=1}^{n}\frac{([Ax]_i - b_i)^2}{[Ax]_i + \sigma^2}\right\} = 1, \tag{13}$$

an obvious choice would be to use Morozov's discrepancy principle [26, 34]. However, as reported in [33], even without outliers, the discrepancy principle based on (13) tends to provide unsatisfactory reconstructions for problems with small signal-to-noise ratio. Therefore, we will not consider this approach further.

3.2 Generalized cross validation

Generalized cross validation [12][34, chap. 7] is a method derived from the standard leave-one-out cross validation. To apply this method for linear Tikhonov regular-

ization, one selects the regularization parameter λ such that it minimizes the GCV functional

$$\mathrm{GCV}(\lambda) = \frac{n\|r_\lambda\|^2}{(\mathrm{trace}(I - A_\lambda))^2}, \tag{14}$$

where $r_\lambda = Ax_\lambda - b = (A_\lambda - I)b$ is the residual, n is its length, and the influence matrix A_λ takes the form $A_\lambda = A(A^TA + \lambda L^TL)^{-1}A^T$. Here, due to the non-negativity constraints and the weights, the residual and the influence matrix have a more complicated form. An approximation of the influence matrix for problems with mixed noise, but without outliers, has been proposed in [1]. There the numerator of the GCV functional takes the form $n\|Wr_\lambda\|^2$ and the approximate influence matrix

$$A_\lambda = WA(D_\lambda(A^TW^2A + \lambda L^TL)D_\lambda)^\dagger D_\lambda A^TW, \tag{15}$$

where W and D_λ are diagonal matrices:

$$W_{ii} = \frac{1}{\sqrt{[Ax_\lambda]_i + \sigma^2}};$$
$$[D_\lambda]_{ii} = \begin{cases} 1, & [x_\lambda]_i > 0, \\ 0, & \text{otherwise}, \end{cases}$$

and † denotes the Moore-Penrose pseudoinverse. Matrix D_λ only handles the non-negativity constraints and, therefore, can be adopted directly. The matrix W needs a special adjustment, due to the change of the loss function to Talwar. The aim is to construct a matrix W satisfying

$$\frac{1}{2}\|Wr_\lambda\|^2 = \sum_{i-1}^{m} \rho\left(\frac{[Ax_\lambda]_i - b_i}{\sqrt{[Ax_\lambda]_i + \sigma^2}}\right).$$

Substituting for ρ from the definition of the function Talwar (9), we redefine the scaling matrix as

$$W_{ii} \equiv \begin{cases} \frac{1}{\sqrt{[Ax_\lambda]_i+\sigma^2}}, & \left|\frac{[Ax_\lambda]_i-b_i}{\sqrt{[Ax_\lambda]_i+\sigma^2}}\right| \le \beta, \\ \frac{\beta}{[Ax_\lambda]_i-b_i}, & \text{otherwise}; \end{cases}$$

In order to make the evaluation of (15) feasible for large-scale problems, we approximate the trace of a matrix using the random trace estimation [19, 34] as $\mathrm{trace}(M) \approx v^TMv$, where the entries of v take values ± 1 with equal probability. Applying the random trace estimation to (15), we obtain

$$(\mathrm{trace}(I - A_\lambda))^2 \approx (v^Tv - v^TA_\lambda v)^2.$$

Finally, $A_\lambda v$ is approximated by WAy, with y obtained applying truncated conjugate gradient iteration to

$$(D_\lambda(A^T W^2 A + \lambda L^T L)D_\lambda)y = D_\lambda A^T W v. \tag{16}$$

4 Minimization problem

In this section, we discuss numerical methods to compute a minimum of (12). We consider incorporating a non-negative constraint and solution of linear subproblems, including proposing a preconditioner.

4.1 Projected Newton's method

Various methods for constrained optimization have been developed over the years; some related to image deblurring can be found in [2, 4, 15, 25, 29]. For our computations, we chose a projected Newton's method,[1] combined with projected PCG to compute the search direction in each step; see [14, sec. 6.4]. The convenience of this method lies in the fact that the projected PCG does not require any special form of the preconditioner and a generic conjugate gradient preconditioner can be employed. Besides lower bounds, upper bounds on the reconstruction can also be enforced. For completeness, we include the projected Newton method in Algorithm 1, and projected PCG in Algorithm 2.

Algorithm 1 Projected Newton's method [14]

input: J_λ, $x^{(0)}$
$k = 0$
while not converged **do**
  Active $= (x^{(k)} \leq 0)$
  $g = \text{grad}_{J_\lambda}(x^{(k)})$
  $H = \text{Hess}_{J_\lambda}(x^{(k)})$
  $M = \texttt{prec}(H)$ setup preconditioner for the Hessian
  $s = \texttt{projPCG}(H, -g, \text{Active}, M)$ compute the search direction for inactive cells
  $g_a = g(\text{Active})$
  if $\max(\text{abs}(g_a)) > \max(\text{abs}(s))$ **then**
    $g_a = g_a \cdot \max(\text{abs}(s))/\max(\text{abs}(g_a))$ rescaling needed
  end if
  $s(\text{Active}) = g_a$ take gradient direction in active cells
  $x^{(k+1)} = \texttt{linesearch}(s, x^{(k)}, J_\lambda, \text{grad}_{J_\lambda})$
  $k = k + 1$
end while
return $x^{(k)}$

[1] In [14], the method was derived as the Projected Gauss–Newton method. Here, since the evaluation of the Hessian does not represent a computational difficulty, we use it as a variant of Newton's method. Therefore, in the remainder of the text, the method is referred to as the Projected Newton's Method.

Algorithm 2 Projected PCG (`projPCG`)[14]

input: A, b, Active, M
$x_0 = 0$
$D_{\mathcal{I}} = \text{diag}(1 - \text{Active})$ projection onto inactive set
$r_0 = D_{\mathcal{I}} b$
$z_0 = D_{\mathcal{I}}(M^{-1} r_0)$
$p_0 = z_0$
$k = 0$
while not converged **do**
  $\alpha_k = \frac{r_k^T z_k}{p^T D_{\mathcal{I}} A p_k}$
  $x_{k+1} = x_k + \alpha_k p_k$
  $r_{k+1} = x_k - \alpha_k D_{\mathcal{I}} A p_k$
  $z_{k+1} = D_{\mathcal{I}}(M^{-1} r_k)$
  $\beta_{k+1} = \frac{z_{k+1}^T r_{k+1}}{z_k^T r_k}$
  $p_{k+1} = z_{k+1} + \beta_k p_k$
  $k = k + 1$
end while
return x_k

4.2 Solution of the linear subproblems

Each step of the projected Newton method (Algorithm 1) requires solving a linear system with the Hessian:

$$\begin{aligned} \text{Hess}_{J_\lambda}(x^{(k)})s &= -\text{grad}_{J_\lambda}(x^{(k)}) \\ (A^T D^{(k)} A + \lambda L^T L)s &= -\left(A^T z^{(k)} + \lambda L^T L x^{(k)}\right). \end{aligned} \tag{17}$$

For the objective functional (12), the diagonal matrix $D^{(k)}$ and the vector $z^{(k)}$ have the form

$$\begin{aligned} z_i &= \begin{cases} \frac{1}{2}\left(1 - \frac{(b_i+\sigma^2)^2}{([Ax]_i+\sigma^2)^2}\right), & \left|\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right| \le \beta, \\ 0, & \text{otherwise.} \end{cases} \\ D_{ii} &= \begin{cases} \frac{(b_i+\sigma^2)^2}{([Ax]_i+\sigma^2)^3}, & \left|\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right| \le \beta, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{18}$$

Note that in case of constant weights, robust regression represents extra computational cost in comparison with standard least squares because it leads to a sequence of weighted least squares problems, while standard least squares problems are solved in one step. In our setting, the weights in (5) themselves have to be updated and therefore employing a different loss function does not change the type of the problem we need to solve.

Without preconditioning, the convergence of projected PCG can be rather slow, and it is therefore important to consider preconditioning. The idea of many preconditioners, such as constraint [8, 20], constraint-type [7], or Hermitian and skew-Hermitian [3] preconditioners is based on the fact that in many cases it is possible to efficiently solve the linear system in (17) if the diagonal matrix $D^{(k)}$ is the identity matrix; that is, if the linear system involves the matrix

$$A^T A + \lambda L^T L. \tag{19}$$

For example, in the case of image deblurring, it is well known that linear systems involving the matrix (19) can be solved efficiently using fast trigonometric or fast Fourier transforms (FFT).

Although the constraint-type, and Hermitian and skew-Hermitian preconditioners seem to perform well for problems with a random matrix $D^{(k)}$ (i.e., a random row scaling), see [3], they performed unsatisfactorily for problems of the form (17), (18).

A preconditioner based on a similar idea of fast computations with matrices of type (19) for imaging problems was proposed in [10]. In this case, the row scaling is approximated by a column scaling; that is, we find $\hat{D}^{(k)}$ such that

$$A^T D^{(k)} A \approx \hat{D}^{(k)} (A^T A) \hat{D}^{(k)}, \tag{20}$$

where

$$\hat{D}^{(k)}_{ii} \equiv \sqrt{\frac{e_i^T (A^T D^{(k)} A) e_i}{e_i^T (A^T A) e_i}}. \tag{21}$$

Note that for $\hat{D}^{(k)}$ defined in (21), the diagonals of the matrices on the two sides of approximation (20) are exactly equal.

Since for large-scale problems matrix A is typically not formed explicitly, exact evaluation of the entries of $\hat{D}^{(k)}$ might become too expensive. To get around this restriction, note that

$$e_i^T (A^T D^{(k)} A) e_i = ((A^T) .^2 \operatorname{diag}(D^{(k)}))_i \quad \text{and} \quad e_i^T (A^T A) e_i = ((A^T) .^2 \mathbf{1})_i, \tag{22}$$

where $\mathbf{1}$ is a vector of all ones, and we use MATLAB notation.^{2} to mean componentwise squaring. In some cases, it may be relatively easy to compute both the entries of $(A^T).^2$ and the vector $(A^T).^2\mathbf{1}$; this is the case for image deblurring and is discussed in more detail in Section 5.

Using (20), we define the preconditioner for the linear system (17) as

$$M \equiv \hat{D}^{(k)} \left(A^T A + \hat{\lambda} L^T L \right) \hat{D}^{(k)}, \tag{23}$$

with

$$\hat{\lambda} \equiv \lambda / \text{mean} \left(\operatorname{diag}(\hat{D}^{(k)}) \right)^2.$$

More details on the computational costs involved in constructing and applying the preconditioner in the case of image deblurring are provided in Section 5.

5 Numerical tests

The Poisson–Gaussian model arises naturally in image applications, so in this section, we present numerical examples from image deblurring. Specifically, we consider the inverse problem (1) with data model (2), where vector b is an observed image that is corrupted by blur and noise, matrix A models the blurring operation, vector x_{true} is the true image, and η is noise. Although an image is naturally represented as an array of pixel values, when we refer to "vector" representations, we assume the pixel values have been reordered as vectors. For example, if we have a $p \times p$ image of pixel values, these can be stored in a vector of length $n = p^2$ by, for example, lexicographical ordering of the pixel values.

In many practical image deblurring applications, the blurring is spatially invariant, and A is structured matrix defined by a *point spread function* (PSF). In this situation, image deblurring can also be referred to as image decovolution, because the operation Ax_{true} is the convolution of x_{true} and the PSF. Although the PSF may be given as an actual function, the more common situation is to compute estimates of it by imaging point source objects. Thus, the PSF can be represented as an image; we typically display the PSF as a mesh plot, which makes it easier to visualize how a point in an image is spread to its neighbors because of the blurring operation. The precise structure of the matrix A depends on the imposed boundary condition; see [17] for details. In this section, we impose periodic boundary conditions, so that A and L are both diagonalizable by FFTs.

So far, we have only described what we refer to as the *single-frame* situation, where b is a single observed image. It is often the case, especially in astronomical imaging, to have multiple observed images of the same object, but with each having a different blurring matrix associated with it. We refer to this as the multi-frame image deblurring problem. Here, b represents all observed images, stacked one on top of each other, and similarly A is formed by stacking the various blurring matrices.

Before describing the test problems used in this section, we first summarize the computational costs. From the discussion around (22), to construct the preconditioner, we need to be able to efficiently square all entries of the matrix A^T, or equivalently those of A; this can easily be approximated by squaring the point-spread function component-wise before forming the operator, i.e.,

$$(A_{\text{PSF}}).^2 \approx A_{\text{PSF}.^2}\,.$$

Using this approximation, in each Newton step, we only need to perform one multiplication by a matrix, one component-wise multiplication, and one component-wise square-root to obtain the entries of the diagonal matrix (21). With the assumption that A and L are both diagonalizable by FFTs, efficient multiplication by the Hessian (17) involves two two-dimensional forward and inverse FFTs, which we refer to as `fft2` and `ifft2`, respectively. Solving systems with matrix (23) involves only one `fft2` and one `ifft2`. In addition to the `fft2` requirements, multiplication by the Hessian (17) involves four pixel-wise multiplications and one addition. Solving systems with the preconditioner (23) involves three pixel-wise multiplications (component-wise reciprocals are assumed to be computed only once at the beginning). The total counts for each operation are shown in Table 1.

Table 1 Operation counts for single-frame case

	Operation	`fft2`	`ifft2`	Mults	Adds
Hessian (17)	Multiply	2	2	4	1
Preconditioner (23)	Solve	1	1	3	0

The robustness and the efficiency of the proposed method is demonstrated on two test problems adopted from [28]:

Satellite An atmospheric seeing problem with spatially invariant atmospheric blur (moderate seeing conditions with the Fried parameter 30). We also consider a multi-frame case, where the same object is blurred by three different PSFs. These PSFs are generated by transposing and flipping the first PSF. The setting is shown in Figs. 2 and 4a.

Carbon ash An image deblurring problem with spatially invariant non-separable Gaussian blur, where the PSF has the functional definition

$$\mathrm{PSF}(s,t) = \frac{1}{2\pi\sqrt{\gamma}} \exp\left\{-\frac{1}{2}\begin{bmatrix} s & t \end{bmatrix} C^{-1} \begin{bmatrix} s \\ t \end{bmatrix}\right\},$$

where

$$C = \begin{bmatrix} \gamma_1^2 & \tau^2 \\ \tau^2 & \gamma_2^2 \end{bmatrix}, \quad \text{and} \quad \gamma_1^2\gamma_2^2 - \tau^4 > 0\,.$$

The shape of the Gaussian PSF depends on the parameters γ_1, γ_2, and τ; we use $\gamma_1 = 4$, $\gamma_2 = 2$; $\tau = 2$. We also consider a multi-frame case, where the same object is blurred by three different PSFs. The other two PSFs are Gaussian blurs with parameters $\gamma_1 = 4$, $\gamma_2 = 2$, $\tau = 0$, and $\gamma_1 = 4$, $\gamma_2 = 2$, $\tau = 0$. The setting is shown in Figs. 3 and 4b.

As previously mentioned, in the multi-frame case, the vector b in (1) is concatenation of the vectorized blurred noisy images; the matrix A is concatenation of the blurring operators, i.e., $A \in \mathbb{R}^{3n\times n}$. For the test problems, all true images are 256×256 arrays of pixels (with intensities scaled to $[0, 255]$), and thus, $n = 65536$.

Computation was performed in MATLAB R2015b. Noise is generated artificially using MATLAB functions `poissrnd` and `randn`. Unless specified otherwise, the

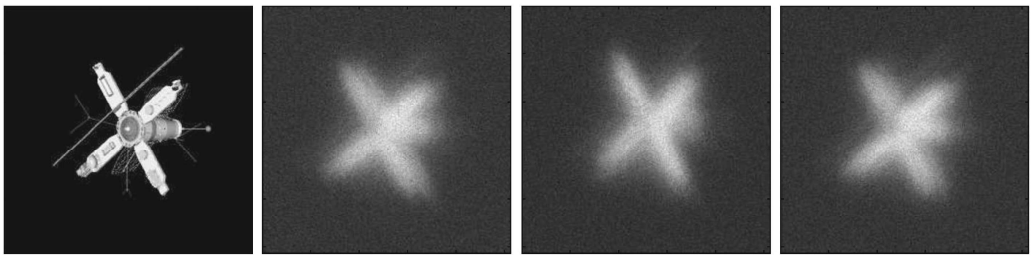


Fig. 2 Test problem Satellite: true image (left) together with three blurred noisy images (right)

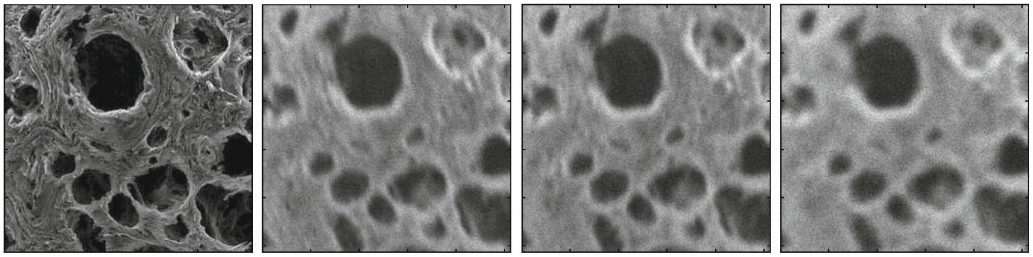


Fig. 3 Test problem Carbon ash: true image (left) together with three blurred noisy images (right)

standard deviation σ is set to 5. We use the discretized Laplacian, see [17, p. 95], as the regularization matrix L. The projected Newton method (Algorithm 1) is terminated when the relative size of the projected gradient

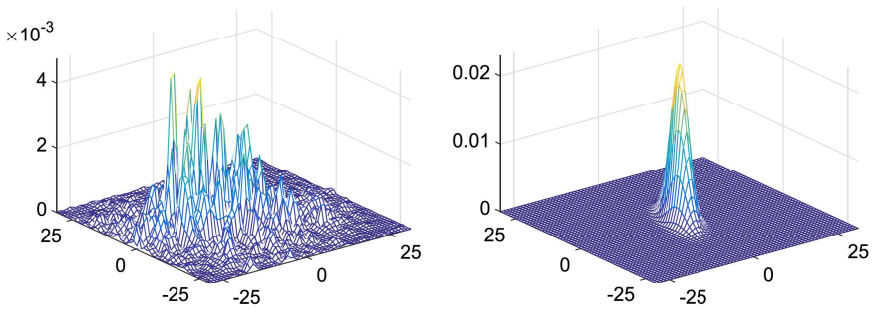
$$\mathcal{P}(\text{grad}_{J_\lambda}(x^{(k)})), \quad \text{where} \quad \mathcal{P}(v) \equiv v. * (1 - \text{Active}) + \text{Active}. * (v < 0),$$

reaches the tolerance 10^{-4} or after 40 iterations. We use MATLAB notation .* to mean component-wise multiplication. Projected PCG (Algorithm 2) is terminated when the relative size of the projected residual (denoted in Algorithm 2 by r_i) reaches 10^{-1}, or the number of iterations reaches 100. We use the preconditioner given in (23) as the default preconditioner. Given a search direction s_k, we apply a projected backtracking linesearch (see, e.g., [2]), with the initial step length equal to 1 and the step-size reduction parameter equal to $1/2$, which we terminate when

$$J_\lambda(x^{(k+1)}) < J_\lambda(x^{(k)}).$$

5.1 Robustness with respect to various types of outliers

In this section, we consider several types of outliers, whose choice was motivated by [5], and demonstrate the robustness of the proposed method. Note that the difference between [5] and the proposed approach lies, among others, in the fact that while



(a) Satellite (b) Carbon ash

Fig. 4 Point-spread functions for the first frame of each test problem

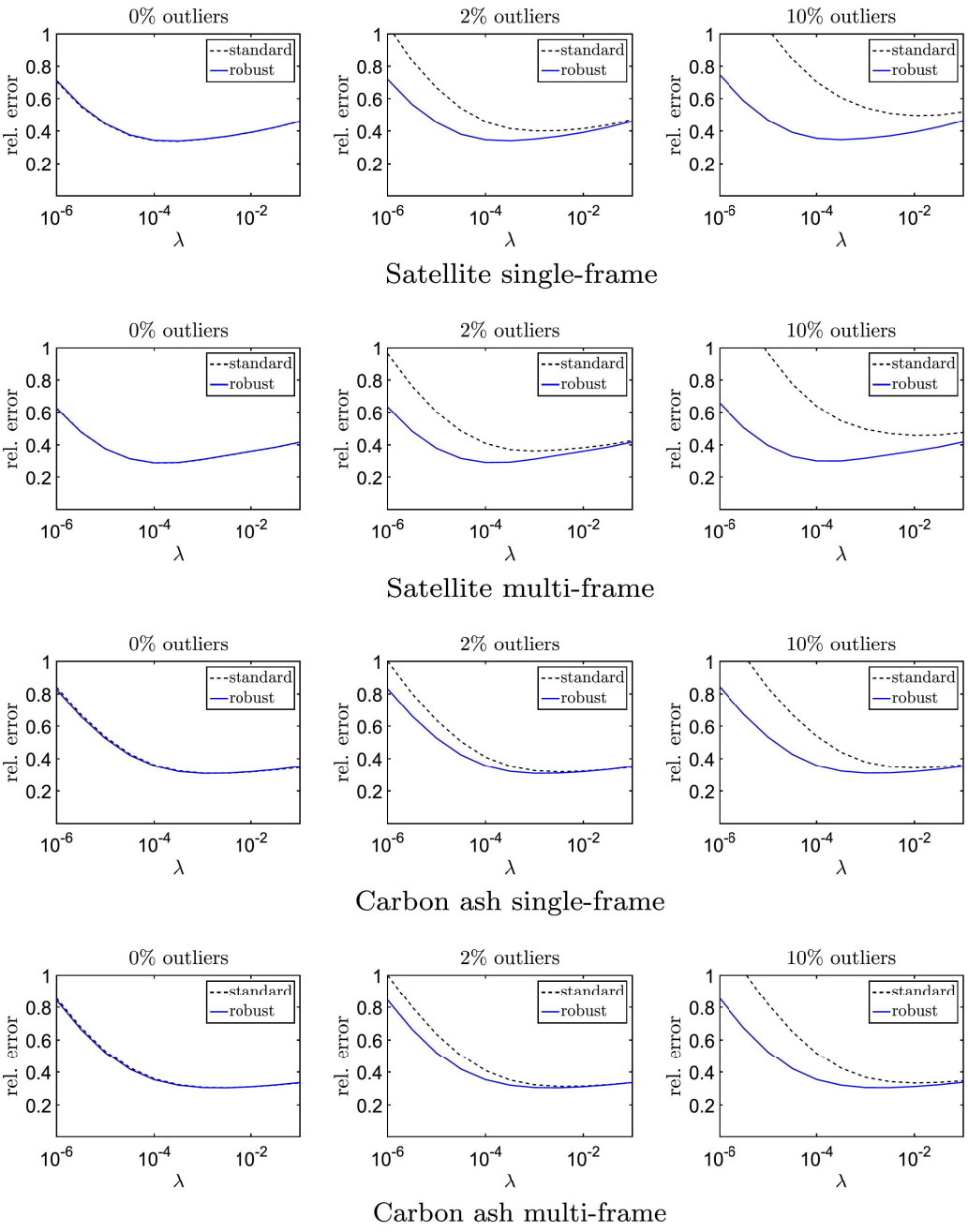
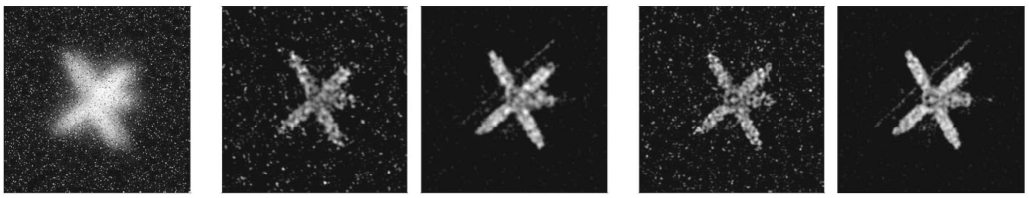


Fig. 5 Semiconvergence curves—dependence of the relative error of the reconstruction on the size of the regularization parameter λ for various percentages of outliers: Talwar (8)–(11) (solid line) and the standard data fidelity function (5) (dashed line)

in [5], the approximation of the solution is computed in order to update the outer (robust) weights associated with the components of residual. Here, the weights are represented by the loss function ρ and are updated implicitly in each Newton step and therefore our approach does not involve any outer iteration.



(a) data (b) single-frame standard (c) single-frame robust (d) multi-frame, standard (e) multi-frame robust

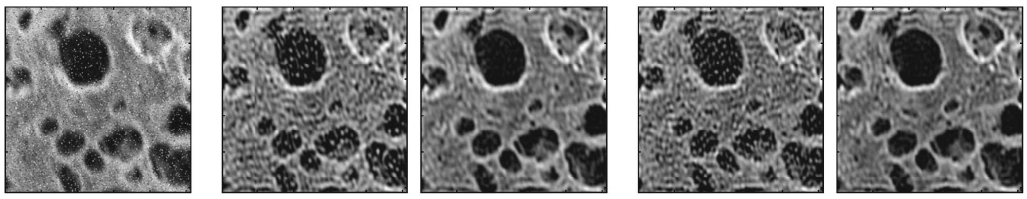
Fig. 6 Random corruptions: **a** blurred noisy image with 10% corrupted pixels (only first frame is shown); **b–e** reconstructions corresponding to $\lambda = 10^{-4}$

5.1.1 Random corruptions

First we consider the most simple case of the outliers—a given percentage of pixels is corrupted at random. These corruptions are generated artificially by adding a value randomly chosen between 0 and $\max(Ax_{\text{true}})$ to the given percentage of pixels. The location of these pixels is also chosen randomly. Figure 5 shows semiconvergence curves,[2] representing the dependence of the error on the regularization parameter λ, when we increase the percentage of corrupted pixels.

It is no surprise that when outliers occur, more regularization is needed in order to obtain a reasonable approximation of the true image x_{true}. This is however not the case if we use loss function Talwar, for which the semiconvergence curve remains the same even with increasing percentage of outliers, and therefore no adjustment of the regularization parameter is needed. In Figs. 6 and 7, we show the reconstructions corresponding to 10% outliers. The regularization parameter is chosen as a close-to-the-optimal regularization parameter for the same problem with no outliers. Note that Figs. 6 and 7 show only one frame for illustration. In the multi-frame case, the corruptions look similar for all frames, except that the random locations of the outliers are different. For random outliers like this, robust regression is clearly superior to standard-weighted least squares. The influence of the outliers in the multi-frame case is less severe, due to intrinsic regularization of the overdetermined system (1). A more comprehensive comparison of the standard and robust approach is shown in Table 2, giving the percentage of cases in which the robust approach provides better reconstruction. The robust approach provides better reconstruction in all cases except for the test problem Satellite with no outliers, where the standard approach gave sometimes slightly better reconstructions. However, even in these cases, we observed that the difference between the errors of the reconstructions is rather negligible, about 3%.

[2]For ill-posed problems, the relative error of an iterative method generally does not decrease monotonically. Instead, unless the problem is highly over-regularized, the relative errors decrease in the early iterations, but at later iterations, the noise and other errors tend to corrupt the approximations. This behavior, where the relative errors decrease to a certain level and then increase at later iterations, is referred to as *semiconvergence*; for more information, we refer readers to [9, 16, 27, 34].



(a) data (b) single-frame standard (c) single-frame robust (d) multi-frame, standard (e) multi-frame robust

Fig. 7 Random corruptions: **a** blurred noisy image with 10% corrupted pixels (only first frame is shown); **b–e** reconstructions corresponding to $\lambda = 10^{-3}$

5.1.2 Added object with different blurring

We also consider a situation when a small object appears in the scene, but is blurred by a different PSF than the main object (satellite). The aim is to recover the main object, while suppressing the influence of the added one. In our case, the added object is a small satellite in the left upper corner that is blurred by a small motion blur. In the multi-frame case, the small satellite is added to the first frame only. The difference between the reconstructions using standard and robust approach is shown in Fig. 8. For the single frame problem, reconstructions obtained using the standard loss function is fully dominated by the small added object. For the multi-frame situation, the influence of the outlier is somewhat compensated by the two frames without outliers. In both cases, however, robust regression provides better reconstruction, comparable to the reconstruction from the data without outliers.

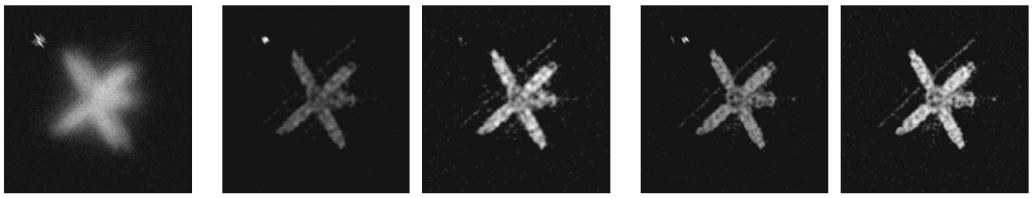
5.1.3 Outliers introduced by boundary conditions

Defining the boundary conditions plays an important role in solving image deblurring problems. As is well known, see e.g. [17], unless some strong a priori information about the scene outside the borders is available, any choice of the boundary conditions may lead to artifacts around edges in the reconstruction. Similarly as in [5], we

Table 2 Comparison of the quality of reconstruction for the standard vs. the robust approach

Better reconstruction: robust/same/standard				
Problem/% outliers	0%	1%	2%	5%
Satellite single-frame	0/93/7	100/0/0	100/0/0	100/0/0
Satellite multi-frame	0/94/6	100/0/0	100/0/0	100/0/0
Carbon ash single-frame	0/100/0	100/0/0	100/0/0	100/0/0
Carbon ash multi-frame	0/100/0	100/0/0	100/0/0	100/0/0

For each test problem and each percentage of outliers, the results are averaged over 100 independent positions and sizes of random corruptions. Regularization parameters are chosen identically as in Figs. 6 and 7. Reconstructions are considered to be of the same quality if the difference between the corresponding relative errors is smaller than 1%



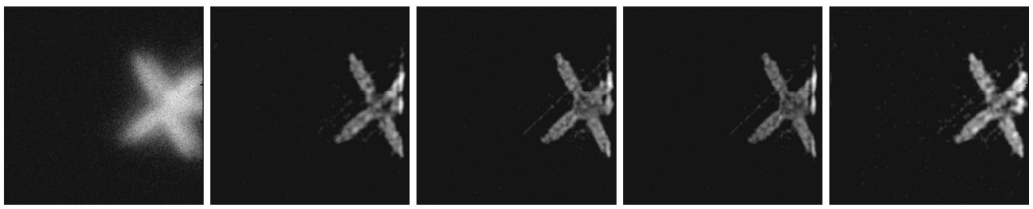
(a) data (b) single-frame standard (c) single-frame robust (d) multi-frame, standard (e) multi-frame robust

Fig. 8 Added object: **a** blurred noisy image with a small object added to the first frame (only first frame is shown); **b–e** reconstructions corresponding to $\lambda = 10^{-4}$

may expect that the robust objective functional (12) can to some extent compensate for these edge artifacts, i.e., the outliers are represented by the "incorrectly" imposed boundary conditions. In our model, we assume periodic boundary conditions, which is computationally very appealing, since it allows evaluating the multiplication by A very efficiently using the fast Fourier transform. However, if any of the objects in the scene are close to the boundary, these boundary conditions will most probably cause artifacts. In order to demonstrate the ability of the proposed scheme to eliminate influence of this type of outlier, we shifted the satellite to the right edge of the image. Other settings remain unchanged. Reconstructions using standard and robust approach are shown in Fig. 9. We see that, although not spectacular, robust regression can reduce the artifacts caused by incorrectly imposed boundary conditions and therefore provide better reconstruction of the true image. Quantitative results for this and all the previous types of outliers are shown in Table 3.

5.2 Generalized cross validation

For the remainder of this section, we will only assume the robust approach, i.e., functional (12) with the loss function Talwar. In Section 3.2, we described a regularization parameter selection rule based on leave-one-out cross validation. Since GCV belongs to standard methods, we focus here mainly on the influence of the outliers on its reliability. To obtain various noise levels, we scale the original true scene



(a) data (b) single-frame standard (c) single-frame robust (d) multi-frame, standard (e) multi-frame robust

Fig. 9 Incorrectly imposed periodic boundary conditions: **a** blurred noisy image close to the edge (only first frame is shown); **b–e** reconstructions corresponding to $\lambda = 10^{-4}$

Table 3 Comparison of the standard and robust approach in terms of relative error of the reconstruction

	Standard		Robust	
Problem	# it	Rel. error	# it	Rel. error
(a) Single-frame				
Satellite	15	3.40×10^{-1}	16	3.42×10^{-1}
Satellite random corr. 10%	14	6.78×10^{-1}	14	3.57×10^{-1}
Carbon ash	10	3.10×10^{-1}	11	3.08×10^{-1}
Carbon ash random corr. 10%	11	3.80×10^{-1}	14	3.10×10^{-1}
Satellite added object	15	4.72×10^{-1}	15	3.43×10^{-1}
Satellite boundary conditions	15	5.48×10^{-1}	25	4.51×10^{-1}
(b) Multi-frame				
Satellite	12	2.89×10^{-1}	11	2.89×10^{-1}
Satellite random corr. 10%	11	6.45×10^{-1}	13	3.00×10^{-1}
Carbon ash	12	3.07×10^{-1}	11	3.05×10^{-1}
Carbon ash random corr. 10%	9	3.70×10^{-1}	19	3.06×10^{-1}
Satellite added object	13	3.33×10^{-1}	11	2.90×10^{-1}
Satellite boundary conditions	14	5.26×10^{-1}	14	4.27×10^{-1}

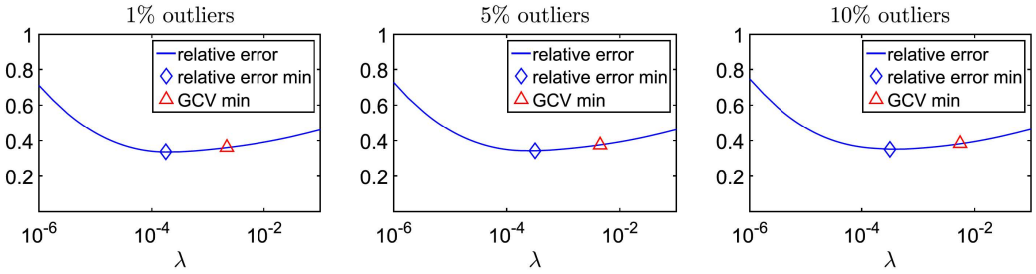
Each row contains results for the standard and robust approach. Abbreviation "# it" stands for the number of Newton steps performed before the relative size of the projected gradient reached the tolerance 10^{-4}. Corresponding reconstructions are shown in Figs. 6, 7, 8, and 9

(with maximum intensity = 255) by 10 and by 100, which results in a decrease of the relative size of Poisson noise. The standard deviation σ for the additive Gaussian noise is scaled accordingly by $\sqrt{10}$ and 10. We compute the resulting signal-to-noise ratio as the reciprocal of the coefficient of variation, i.e.,

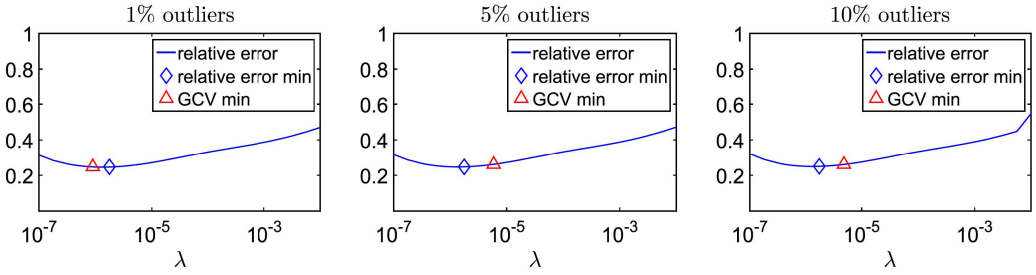
$$\mathrm{SNR} = \frac{\|Ax\|}{\sqrt{\sum_{i=1}^{n}([Ax]_i + \sigma^2)}}.$$

For our computations, we use CG to solve (16), which we terminate if the relative size of the residual reaches 10^{-4} or if the number of iterations reaches 150. To minimize the GCV functional, we use the MATLAB built-in function `fminbnd`, for which we set the lower bound to 0 and the upper bound to 10^{-1}, 10^{-2}, 10^{-4}, depending on the maximum intensity of the image. The tolerance `TolX` was set to 10^{-8}.

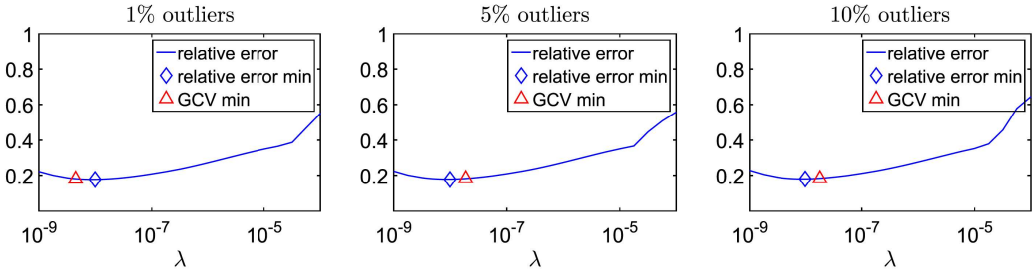
For test problem Satellite, we show the semiconvergence curves including the minimum error and the error obtained using GCV in Fig. 10. Quantitative results (averaged over 10 realizations of outliers) for both test problems are shown in Table 4. We observe that the proposed rule is rather stable with respect to the increasing number of outliers and generally better for the Carbon ash than for the Satellite. As expected, the method provides better approximation of the optimal regularization



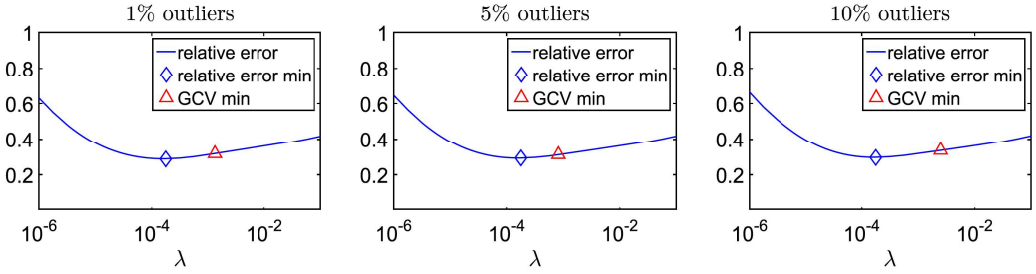
Satellite single-frame, max. intensity 255 (SNR = 5).



Satellite single-frame, max. intensity 2550 (SNR = 17).



Satellite single-frame, max. intensity 25500 (SNR = 52).



Satellite multi-frame, max. intensity 255 (SNR = 5).

Fig. 10 GCV for data with outliers

parameter for smaller noise levels (larger Ax_{true}), where the functional (5) approximates better the maximum likelihood functional for the mixed Poisson–Gaussian model. Occasionally, GCV provides slightly worse reconstruction for the highest percentage (10%) of outliers.

Table 4 Relative errors of the reconstruction: optimal λ vs. λ obtained by minimizing the GCV functional (14)

			Error min Error GCV	(λ optimal) (λ GCV)		
		% outliers				
Problem	Max. int.	1%	5%		10%	
(a) Satellite						
Single-frame	255	3.39×10^{-1}	$(\lambda_{opt} = 2.1 \times 10^{-4})$	3.43×10^{-1}	$(\lambda_{opt} = 2.5 \times 10^{-4})$	3.49×10^{-1}	$(\lambda_{opt} = 3.2 \times 10^{-4})$
		3.60×10^{-1}	$(\lambda_{GCV} = 2.1 \times 10^{-3})$	3.68×10^{-1}	$(\lambda_{GCV} = 3.1 \times 10^{-3})$	3.80×10^{-1}	$(\lambda_{GCV} = 5.6 \times 10^{-3})$
Single-frame	2550	2.46×10^{-1}	$(\lambda_{opt} = 1.5 \times 10^{-6})$	2.48×10^{-1}	$(\lambda_{opt} = 1.5 \times 10^{-6})$	2.50×10^{-1}	$(\lambda_{opt} = 1.7 \times 10^{-6})$
		2.48×10^{-1}	$(\lambda_{GCV} = 1.5 \times 10^{-6})$	2.57×10^{-1}	$(\lambda_{GCV} = 4.5 \times 10^{-6})$	2.69×10^{-1}	$(\lambda_{GCV} = 8.4 \times 10^{-6})$
Single-frame	25500	1.78×10^{-1}	$(\lambda_{opt} = 1.0 \times 10^{-8})$	1.79×10^{-1}	$(\lambda_{opt} = 1.0 \times 10^{-8})$	1.81×10^{-1}	$(\lambda_{opt} = 1.0 \times 10^{-8})$
		1.79×10^{-1}	$(\lambda_{GCV} = 8.8 \times 10^{-9})$	1.81×10^{-1}	$(\lambda_{GCV} = 1.5 \times 10^{-8})$	2.42×10^{-1}	$(\lambda_{GCV} = 7.5 \times 10^{-6})$
Multi-frame	255	2.89×10^{-1}	$(\lambda_{opt} = 1.8 \times 10^{-4})$	2.92×10^{-1}	$(\lambda_{opt} = 1.8 \times 10^{-4})$	2.97×10^{-1}	$(\lambda_{opt} = 1.8 \times 10^{-4})$
		3.16×10^{-1}	$(\lambda_{GCV} = 1.6 \times 10^{-3})$	3.13×10^{-1}	$(\lambda_{GCV} = 1.2 \times 10^{-3})$	3.49×10^{-1}	$(\lambda_{GCV} = 5.8 \times 10^{-3})$
(b) Carbon ash							
Single-frame	255	3.08×10^{-1}	$(\lambda_{opt} = 1.8 \times 10^{-3})$	3.09×10^{-1}	$(\lambda_{opt} = 1.8 \times 10^{-3})$	3.10×10^{-1}	$(\lambda_{opt} = 1.7 \times 10^{-3})$
		3.12×10^{-1}	$(\lambda_{GCV} = 8.9 \times 10^{-4})$	3.11×10^{-1}	$(\lambda_{GCV} = 1.5 \times 10^{-3})$	3.11×10^{-1}	$(\lambda_{GCV} = 2.2 \times 10^{-3})$
Single-frame	2550	2.91×10^{-1}	$(\lambda_{opt} = 1.9 \times 10^{-5})$	2.92×10^{-1}	$(\lambda_{opt} = 2.1 \times 10^{-5})$	2.92×10^{-1}	$(\lambda_{opt} = 1.9 \times 10^{-5})$
		2.97×10^{-1}	$(\lambda_{GCV} = 9.7 \times 10^{-6})$	2.95×10^{-1}	$(\lambda_{GCV} = 1.2 \times 10^{-5})$	2.93×10^{-1}	$(\lambda_{GCV} = 2.2 \times 10^{-5})$
Single-frame	25500	2.77×10^{-1}	$(\lambda_{opt} = 3.0 \times 10^{-7})$	2.78×10^{-1}	$(\lambda_{opt} = 2.7 \times 10^{-7})$	2.79×10^{-1}	$(\lambda_{opt} = 2.6 \times 10^{-7})$
		3.00×10^{-1}	$(\lambda_{GCV} = 5.2 \times 10^{-8})$	2.84×10^{-1}	$(\lambda_{GCV} = 9.4 \times 10^{-8})$	2.82×10^{-1}	$(\lambda_{GCV} = 1.3 \times 10^{-7})$
Multi-frame	255	3.03×10^{-1}	$(\lambda_{opt} = 1.8 \times 10^{-3})$	3.04×10^{-1}	$(\lambda_{opt} = 1.8 \times 10^{-3})$	3.04×10^{-1}	$(\lambda_{opt} = 1.8 \times 10^{-3})$
		3.14×10^{-1}	$(\lambda_{GCV} = 6.8 \times 10^{-4})$	3.07×10^{-1}	$(\lambda_{GCV} = 9.8 \times 10^{-4})$	3.05×10^{-1}	$(\lambda_{GCV} = 1.9 \times 10^{-3})$

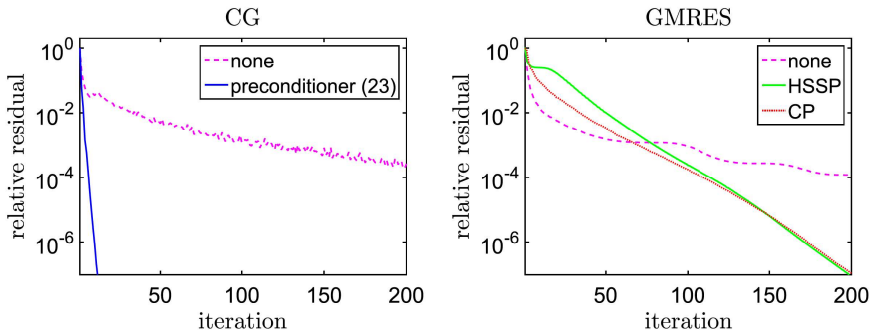


Fig. 11 Preconditoner defined in (23), constraint preconditioner (CP), and Hermitian and skew-Hermitian splitting preconditioner (HSSP) performance for $(A^T DA+\lambda L^T L)s = -A^T b$, where A and b are adopted from the single-frame test problem Satellite, and D is a diagonal with random entries uniformly distributed in $(0, 1)$

5.3 Linear subproblems

As mentioned earlier, various types of preconditioners have been developed to speed up convergence of iterative methods applied to systems of type (17) or its saddle-point counterpart

$$\begin{pmatrix} D^{-1} & A \\ A^T & \lambda L^T L \end{pmatrix} = \begin{pmatrix} -z \\ \lambda L^T Lx \end{pmatrix}.$$

The Hermitian and skew-Hermitian (HSS) preconditioners, as well as the constraint precondtioner, belong to the best known preconditioners for this type of linear system. Both were incorporated in GMRES and tested on deblurring problems with random diagonal scaling D in [3]. Using random D, they indeed accelerate convergence also in our case, as shown in Fig. 11. However, our preconditioner (23) provides a much better speedup. Moreover, for real computations, e.g., when the matrix D is actually generated during the projected Newton computation, the HSS and constraint preconditioners did not perform well and even slowed down the convergence; see Fig. 12. This is fortunately not the case for our proposed preconditioner. In this experiment, we did not assume projection on the non-negative half-plane and

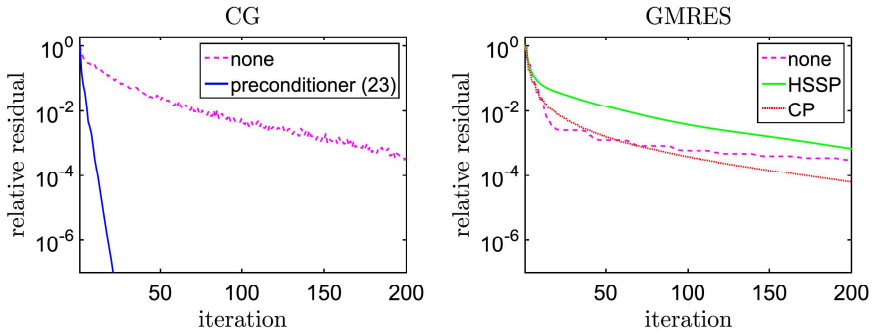
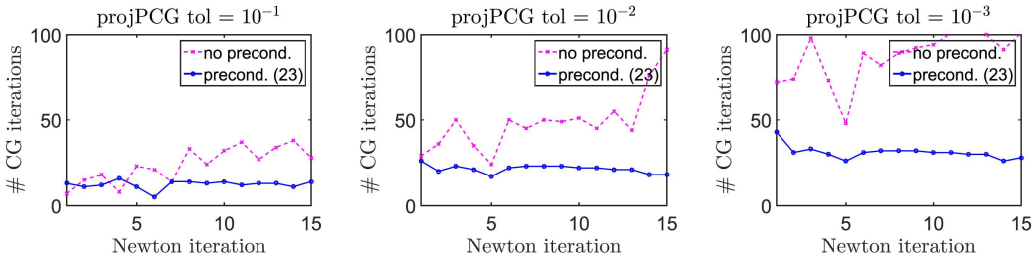
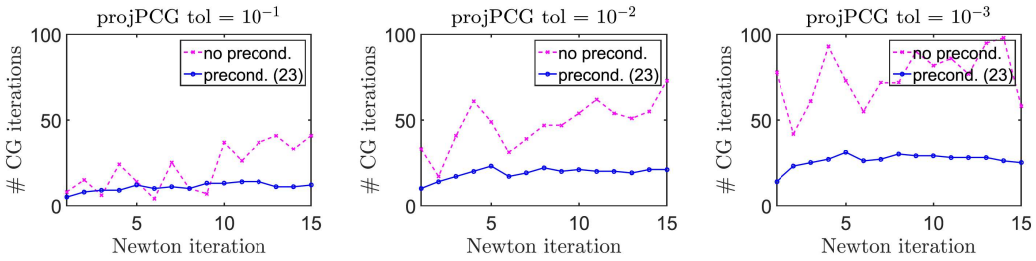


Fig. 12 Preconditoner defined in (23), constraint preconditioner (CP), and Hermitian/skew-Hermitian preconditioner (HSSP) performance for $(A^T D^{(k)} A+\lambda L^T L)s = -\left(A^T z^{(k)} + \lambda L^T Lx^{(k)}\right)$, with $D^{(k)}$, $z^{(k)}$, and $x^{(k)}$ adopted from the third iteration of Newton method for the single-frame test problem Satellite

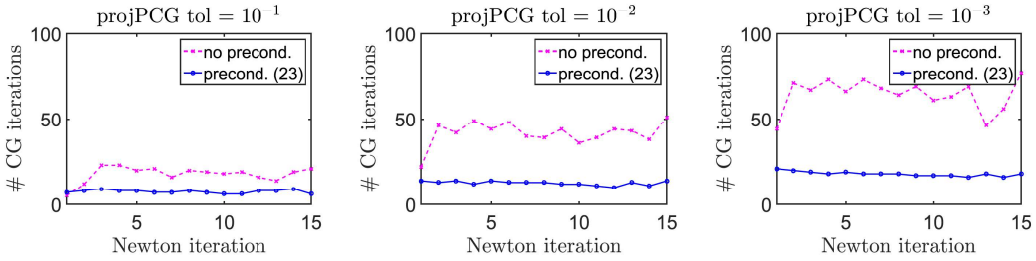
since in (5.3), we need to evaluate D^{-1}, if some component $D_{ii} = 0$, we replaced it by $2\sqrt{\epsilon_{\text{mach}}}$; see also [24]. We also did not incorporate any outliers for these initial experiments with the preconditioners; these results are intended to show that our proposed preconditioning for these problems often performs much better than the well-known standard preconditioners. In fact, we see that the behavior of the constraint and HSS preconditioner depends heavily on the actual setting of the problem.



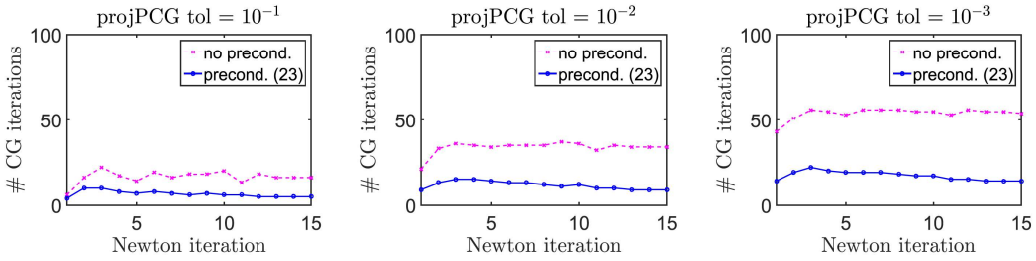
(a) Satellite single-frame



(b) Satellite multi-frame



(c) Carbon Ash single-frame



(d) Carbon Ash multi-frame

Fig. 13 The effect of preconditioning by preconditioner defined in (23): number of projPCG iterations performed in each Newton iteration to achieve the desired tolerance. 5 % outliers.

In the remainder of this section, we will therefore focus on the preconditioner given in (23).

In Fig. 13, we investigate the overall speedup of the convergence by plotting the number of projected PCG steps needed in each Newton iteration to reach the desired tolerance on the relative size of the projected gradient. Even for the most generous tolerance 10^{-1}, preconditioner (23) significantly reduces the number of projPCG iterations. Note that in this experiment, the linear subproblems solved in each Newton iteration are generally not identical, since the subproblems are not solved exactly and therefore the approximations $x^{(k)}$ are not the same. We set the outer tolerance to 0 in order to perform always at least 15 Newton iterations.

The choice of projPCG tolerance is a difficult question, but from the average number of Newton iterations/projPCG iterations/fast Fourier transforms shown in Table 5, we observe that raising the tolerance does not considerably increase the number of Newton steps we need to perform here. Therefore, larger tolerance, here 10^{-1}, leads to a smaller total number of projPCG iterations. This is independent of the percentage

Table 5 Average number of Newton iterations, projPCG iterations, and (inverse) 2D Fourier transforms for projPCG with and without preconditioning, and two tolerances on the relative size of the projPCG residual. Results are averaged over 10 independent realizations of noise and outliers

Average count: Newton/CG/`fft2s`				
		% outliers		
Problem	Precond	0%	2%	10%
(a) projPCG tol = 10^{-1}				
Satellite single-frame	No	14/290/1383	14/274/1329	14/283/1374
	Yes	15/161/1280	16/172/1362	14/158/1252
Satellite multi frame	No	12/250/2398	12/216/2104	13/241/2364
	Yes	12/107/1545	11/107/1535	12/103/1507
Carbon ash single-frame	No	11/190/939	10/179/891	11/184/915
	Yes	10/71/641	11/72/654	13/82/753
Carbon ash multi-frame	No	14/221/2200	14/219/2179	16/254/2542
	Yes	16/88/1510	15/85/1419	17/99/1654
(b) projPCG tol = 10^{-2}				
Satellite single-frame	No	13/536/2359	13/539/2373	14/641/2819
	Yes	14/284/2001	15/302/2117	14/296/2091
Satellite multi-frame	No	11/457/4082	11/460/4130	12/499/4511
	Yes	11/201/2536	11/197/2499	12/213/2747
Carbon ash single-frame	No	11/393/1754	11/432/1912	13/526/2315
	Yes	10/121/934	11/129/1004	13/155/1201
Carbon ash multi-frame	No	13/426/3813	14/461/4127	16/498/4467
	Yes	13/144/1954	14/150/2038	16/173/2359

of outliers. For each setting, the number of projPCG iterations is significantly smaller for the preconditioned version. This is not always the case for the total count of the fast Fourier transforms, since we need to perform 6 `fft2`/`ifft2` in each iteration vs. 4 for the unpreconditioned iterations; see Table 1. For large scale problems, however, the computational complexity of fast Fourier transform, which is $\mathcal{O}(n \log n)$ is comparable to other operations performed in projPCG, such as the inner products, whose complexity is $\mathcal{O}(n)$, and therefore the number of projPCG iterations seems to be the more important indicator of efficiency of the preconditioner. Recall here that n is the number of pixels in the image, so if we have a 256×256 array of pixels, then $n = 65535$.

6 Conclusion

We have presented an efficient approach to compute approximate solutions of a linear inverse problem that is contaminated with mixed Poisson–Gaussian noise, and when there are outliers in the measured data. We investigated the convexity properties of various robust regression functions and found that the Talwar function was the best option. We proposed a preconditioner and illustrated that it was more effective than other standard preconditioning approaches on the types of problems studied in this paper. Moreover, we showed that a variant of the GCV method can perform well in estimating regularization parameters in robust regression. A detailed discussion of computational costs, and extensive numerical experiments illustrate the approach proposed in this paper is effective and efficient on image deblurring problems.

Acknowledgements The authors would like to thank Lars Ruthotto for pointing them to the Projected PCG algorithm and providing them with a Matlab code. The first author would like to thank Emory University for the hospitality offered in academic year 2014-2015, when part of this work was completed. We also thank two anonymous referees for their useful comments that significantly improved the presentation of the paper.

References

1. Bardsley, J.M., Goldes, J.: Regularization parameter selection methods for ill-posed Poisson maximum likelihood estimation. Inverse Prob. **25**(9), 095,005 (2009). https://doi.org/10.1088/0266-5611/25/9/095005
2. Bardsley, J.M., Vogel, C.R.: A nonnegatively constrained convex programming method for image reconstruction. SIAM J. Sci. Comput. **25**(4), 1326–1343 (2003/04). https://doi.org/10.1137/S1064827502410451
3. Benzi, M., Ng, M.K.: Preconditioned iterative methods for weighted Toeplitz least squares problems. SIAM J. Matrix Anal. Appl. **27**(4), 1106–1124 (2006). https://doi.org/10.1137/040616048
4. Bonettini, S., Zanella, R., Zanni, L.: A scaled gradient projection method for constrained image deblurring. Inverse Prob. **25**(1), 015,002 (2009). https://doi.org/10.1088/0266-5611/25/1/015002
5. Calef, B.: Iteratively reweighted blind deconvolution. In: 2013 IEEE International Conference on Image Processing, pp. 1391–1393 (2013). https://doi.org/10.1109/ICIP.2013.6738286
6. Coleman, D., Holland, P., Kaden, N., Klema, V., Peters, S.C.: A system of subroutines for iteratively reweighted least squares computations. ACM Trans. Math. Softw. (TOMS) **6**(3), 327–336 (1980). https://doi.org/10.3386/w0189

7. Dollar, H.S.: Constraint-style preconditioners for regularized saddle point problems. SIAM J. Matrix Anal. Appl **29**(2), 672–684 (2007). https://doi.org/10.1137/050626168
8. Dollar, H.S., Gould, N.I.M., Schilders, W.H.A., Wathen, A.J.: Using constraint preconditioners with regularized saddle-point problems. Comput. Optim. Appl. **36**(2-3), 249–270 (2007). https://doi.org/10.1007/s10589-006-9004-x
9. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems Mathematics and Its Applications, vol. 375. Kluwer Academic Publishers Group, Dordrecht (2000)
10. Fessler, J.A., Booth, S.D.: Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction. IEEE Trans. Image Process **8**(5), 688–699 (1999). https://doi.org/10.1109/83.760336
11. Fisher, R.A.: On the mathematical foundations of theoretical statistics. Philos. Trans. R. Soc. Lond. Ser. A, Containing Papers of a Mathematical or Physical Character **222**, 309–368 (1922)
12. Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics **21**(2), 215–223 (1979). https://doi.org/10.2307/1268518
13. Gravel, P., Beaudoin, G., Guise, J.A.D.: A method for modeling noise in medical images. IEEE Trans. Med. Imaging **23**(10), 1221–1232 (2004). https://doi.org/10.1109/TMI.2004.832656
14. Haber, E.: Computational methods in geophysical electromagnetics. Mathematics in Industry SIAM (2015)
15. Hanke, M., Nagy, J.G., Vogel, C.: Quasi-Newton approach to nonnegative image restorations. Linear Algebra Appl **316**(1-3), 223–236 (2000). https://doi.org/10.1016/S0024-3795(00)00116-6. Conference Celebrating the 60th Birthday of Robert J. Plemmons (Winston-Salem, NC, 1999)
16. Hansen, P.: Discrete inverse problems. Society for industrial and applied mathematics (2010)
17. Hansen, P.C., Nagy, J.G., O'Leary, D.P.: Deblurring Images, Fundamentals of Algorithms, Vol. 3. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2006). Matrices, spectra, and filtering
18. Hansen, P.C., Pereyra, V., Scherer, G.: Least Squares Data Fitting with Applications. Johns Hopkins University Press, Baltimore (2013)
19. Hutchinson, M.F.: A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. Communications in Statistics – Simulation and Computation **19**(2), 433–450 (1990). https://doi.org/10.1080/03610919008812864
20. Keller, C., Gould, N.I.M., Wathen, A.J.: Constraint preconditioning for indefinite linear systems. SIAM J. Matrix Anal. Appl. **21**(4), 1300–1317 (2000). https://doi.org/10.1137/S0895479899351805
21. Kubínová, M., Nagy, J.G.: Iteratively Reweighted Deconvolution and Robust Regression. In: 16Th AMOSTECH Conference Proceedings (2015)
22. Luisier, F., Blu, T., Unser, M.: Image denoising in mixed Poisson-Gaussian noise. IEEE Trans. Image Process. **20**(3), 696–708 (2011). https://doi.org/10.1109/TIP.2010.2073477
23. Mäkitalo, M., Foi, A.: Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. IEEE Trans. Image Process. **22**(1), 91–103 (2013). https://doi.org/10.1109/TIP.2012.2202675
24. Mastronardi, N., O'Leary, D.P.: Fast robust regression algorithms for problems with Toeplitz structure. Comput. Stat. Data Anal. **52**(2), 1119–1131 (2008). https://doi.org/10.1016/j.csda.2007.05.008
25. Moré, J.J., Toraldo, G.: On the solution of large quadratic programming problems with bound constraints. SIAM J. Optim **1**(1), 93–113 (1991). https://doi.org/10.1137/0801008
26. Morozov, V.A.: On the solution of functional equations by the method of regularization. Sov. Math. Dokl. **7**, 414–417 (1966)
27. Mueller, J.L., Siltanen, S.: Linear and Nonlinear Inverse Problems with Practical Applications, Computational Science & Engineering, vol. 10. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2012)
28. Nagy, J.G., Palmer, K., Perrone, L.: Iterative methods for image deblurring: a Matlab object-oriented approach. Numerical Algorithms **36**(1), 73–93 (2004). https://doi.org/10.1023/B:NUMA.0000027762.08431.64
29. Nagy, J.G., Strakoš, Z.: Enforcing nonnegativity in image reconstruction algorithms. In: Proceedings of SPIE, vol. 4121, pp. 182–190 (2000). https://doi.org/10.1117/12.402439. http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=918135
30. O'Leary, D.P.: Robust regression computation using iteratively reweighted least squares. SIAM J. Matrix Anal. Appl **11**(3), 466–480 (1990). https://doi.org/10.1137/0611032. Sparse matrices (Gleneden Beach, OR, 1989)
31. Serfling, R.: Asymptotic Relative Efficiency in Estimation, pp. 68–72. Springer, Berlin (2011)

32. Snyder, D.L., White, R.L., Hammoud, A.M.: Image recovery from data acquired with a charge-coupled-device camera. J. Opt. Soc. Am. A **10**(5), 1014–1023 (1993). https://doi.org/10.1364/JOSAA.10.001014
33. Staglianò, A., Boccacci, P., Bertero, M.: Analysis of an approximate model for Poisson data reconstruction and a related discrepancy principle. Inverse Prob. **27**(12), 125,003 (2011). https://doi.org/10.1088/0266-5611/27/12/125003
34. Vogel, C.R.: Computational methods for inverse problems. In: Frontiers in Applied Mathematics, vol. 23. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2002). https://doi.org/10.1137/1.9780898717570. With a foreword by H. T. Banks
35. Xiao, Y., Zeng, T., Yu, J., Ng, M.K.: Restoration of images corrupted by mixed Gaussian impulsive noise via ℓ_1-ℓ_0 minimization. Pattern Recogn. **44**(7), 1708–1720 (2011). https://doi.org/10.1016/j.patcog.2011.02.002
36. Yan, M.: Restoration of images corrupted by impulse noise and mixed gaussian impulse noise using blind inpainting. SIAM J. Imaging Sci. **6**(3), 1227–1245 (2013). https://doi.org/10.1137/12087178X

5.2 A comment on the Gauss–Newton method

To handle the mixed Poisson–Gaussian model with outliers in the data, we assume a combination of a robust loss function with the weights depending on the computed data. The solution-dependent weights, rescaling the variance of the residual components, however constrain the choice of the loss function ρ. Assume as in the previous section the following objective functional

$$J(x) \equiv \sum_{i=1}^{m} \rho\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right), \tag{5.1}$$

where $\rho : \mathbb{R} \mapsto \mathbb{R}_0^+$, see O'Leary [1990]; Coleman et al. [1980] or [Hansen et al., 2013, sec. 1.5]. Note that for simplicity, we do not include the regularization term in (5.1) as well as in the analysis in the remainder of the section. In practical computations, some form of regularization is always necessary and the regularization term can be incorporated to the functional similarly to [Kubínová and Nagy, in press, sec. 3]. As analyzed in [Kubínová and Nagy, in press, sec. 2.1], the Hessian of the functional (5.1) has the form $\mathrm{Hess}_J(x) = A^T D A$, where D is a diagonal matrix with the entries

$$\begin{aligned} D_{ii} = &\frac{\left(\frac{1}{2}[Ax]_i + \frac{1}{2}b_i + \sigma^2\right)^2}{([Ax]_i + \sigma^2)^3} \rho''\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right) \\ &- \frac{\left(\frac{1}{4}[Ax]_i + \frac{3}{4}b_i + \sigma^2\right)}{([Ax]_i + \sigma^2)^{5/2}} \rho'\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right). \end{aligned} \tag{5.2}$$

Note that while the first addend in the right-hand side of (5.2) is always non-negative, provided $\rho'' \geq 0$, the second addend can have either sign. The analysis in [Kubínová and Nagy, in press, sec. 2.1] shows that only the loss function *Talwar* guarantees the non-negativity of the diagonal entries of D and hence the positive-semidefiniteness of the Hessian required for Newton's method.

5.2.1 Gauss-Newton method for robust regression

Considering the standard loss function $\rho(t) = t^2/2$, (5.1) becomes

$$\hat{J}(x) \equiv \frac{1}{2} \sum_{i=1}^{m} \left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right)^2,$$

see also [Kubínová and Nagy, in press, sec. 1.1]. Minimization of this functional leads to a non-linear least-squares problem. For this type of problems, Gauss–Newton method [Nocedal and Wright, 1999, sec. 10.3] is often the method of choice.

Recall that Gauss–Newton method can be viewed as an approximation to Newton's method for non-linear least-squares, with the Hessian substituted by the

product of Jacobians. In our case,

$$\mathrm{Hess}_{\hat{J}}(x) = A^T \hat{D} A \approx A^T \hat{Z}^2 A,$$

where $\hat{Z}$ is diagonal with the diagonal entries

$$\hat{Z}_{ii} = \frac{\frac{1}{2}[Ax]_i + \frac{1}{2}b_i + \sigma^2}{([Ax]_i + \sigma^2)^{3/2}}.$$

[Haber, 2015, sec. 6.4.2] suggests that such approximation can be applied not only to non-linear least-squares problems, but also to functionals involving robust loss functions ρ. Following this idea, we derive a version of the Gauss-Newton method that approximates the Hessian as

$$\mathrm{Hess}_J(x) = A^T D A \approx A^T \tilde{D} A, \quad \text{where } \tilde{D} \equiv \hat{Z}\,\mathrm{diag}\left[\rho''\left(\frac{[Ax]_i - b_i}{\sqrt{[Ax]_i + \sigma^2}}\right)\right]\hat{Z}.$$

Note that the matrix $\tilde{D}$ is diagonal and its entries coincide with the first addend of the right-hand side in (5.2). The search direction is defined in the standard way as a solution to

$$\left(A^T \tilde{D} A\right) s = -\mathrm{grad}_J(x). \tag{5.3}$$

Direction s will be a descent direction as long as $\tilde{D}_{ii}$ is non-negative, i.e., as long as $\rho'' \geq 0$. This is satisfied for loss functions *Talwar*, *Huber*, *Fair*, and *logistic*, see also O'Leary [1990] or [Kubínová and Nagy, in press, Fig. 1]. For the sake of completeness, we summarize in Table 5.1 the first and the second derivatives for these four loss functions, needed to evaluate the diagonal matrix $\tilde{D}$ and the gradient $\mathrm{grad}_J(x)$, together with the tuning parameters β corresponding to the 95% asymptotic efficiency, see [Kubínová and Nagy, in press, sec. 2.2].

Table 5.1: Four robust loss functions ρ with $\rho'' \geq 0$, their derivatives, and 95% asymptotic efficiency tuning parameters.

Function		$\rho(t)$	$\rho'(t)$	$\rho''(t)$	β_{95}
Talwar	$\lvert t\rvert \leq \beta$	$t^2/2$	t	1	2.795
	$\lvert t\rvert > \beta$	$\beta^2/2$	0	0	
Huber	$\lvert t\rvert \leq \beta$	$t^2/2$	t	1	1.345
	$\lvert t\rvert > \beta$	$\beta\lvert t\rvert - \beta^2/2$	$\beta\ \mathrm{sign}(t)$	0	
Fair		$\beta^2(\lvert t\rvert/\beta - \log(1 + \lvert t\rvert/\beta))$	$\frac{\beta t}{\lvert t\rvert + \beta}$	$\frac{\beta^2}{(\lvert t\rvert + \beta)^2}$	1.400
Logistic		$\beta^2 \log(\cosh(t/\beta))$	$\beta \tanh(t/\beta)$	$\mathrm{sech}^2(t/\beta)$	1.205

Applying Gauss-Newton instead of Newton's method will allow us to use three more loss functions ρ to possibly capture better the characteristics of the outliers, while still getting a descent direction in each step. However, substituting the Hessian by its approximation here does not reduce the cost of computing the search

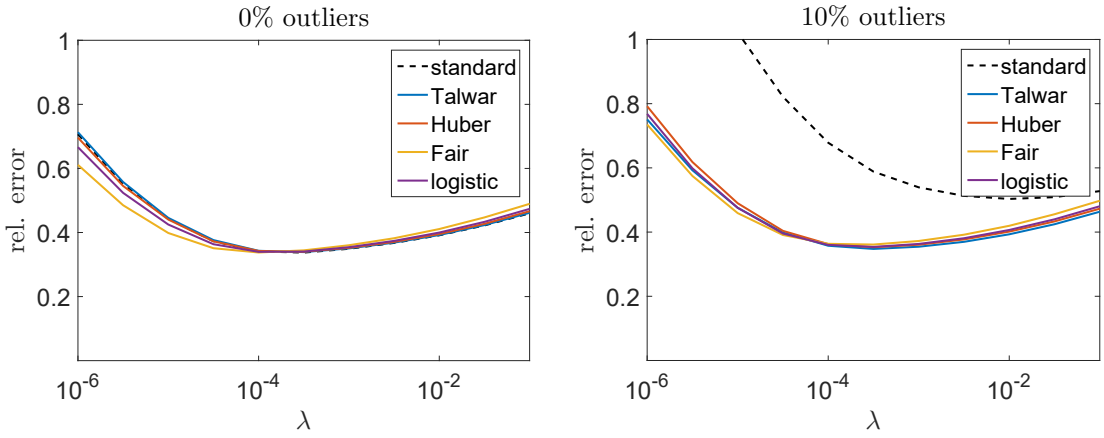
direction s in (5.3), as the structure of the matrix involved in the linear system (5.3) remains the same as for the Newton's method considered in Kubínová and Nagy [in press]. On the other side, the identical structure of the linear problems allows us to use the same optimization scheme including the preconditioner for the linear solves.

5.2.2 Numerical experiments

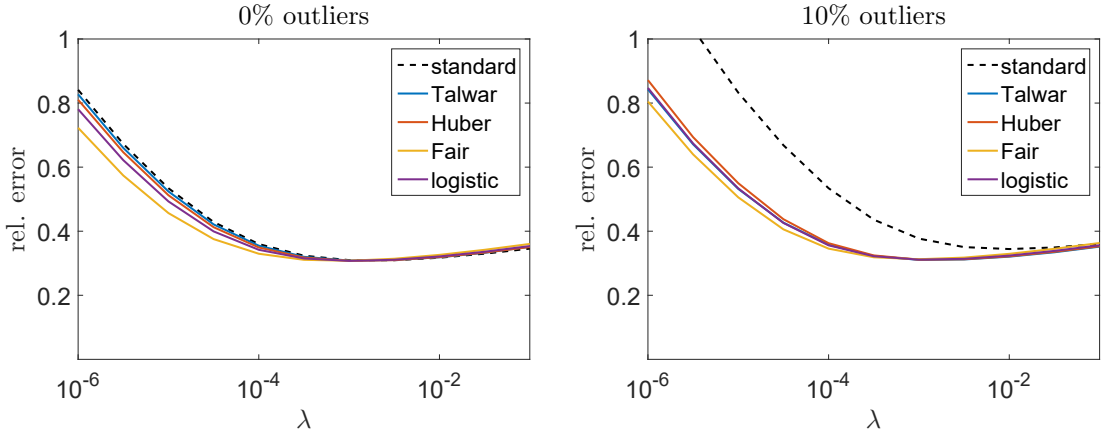
In this section we present numerical experiments involving loss functions Huber, Fair, and logistic. In the first experiment, we focus on the minimum attainable error of reconstruction and the stability of the semiconvergence curves when increasing the percentage of outliers. We performed experiment analogous to experiment described in [Kubínová and Nagy, in press, sec. 5.1.1] and computed the semiconvergence curves for the standard loss function and the four robust loss functions from Table 5.1. For functions Huber, Fair, and logistic, the solution is computed using the Gauss-Newton method as derived above, other settings remain the same as in [Kubínová and Nagy, in press, sec. 5.1.1]. Results for single-frame case are shown in Figure 5.1.

We observe that the semiconvergence curves for all four robust functions ρ show similar behavior, especially around the point of semiconvergence, i.e., the optimal regularization parameter λ. The semiconvergence curves also show similar stability with respect to the increased number of outliers. This remains valid also for multi-frame case (not presented here). In conclusion, for the considered type of problems, the change of the loss function itself does not improve the minimum attainable error significantly, neither in the case when outliers are present, nor in the cases with no outliers. However, we expect that there are cases in image deblurring, where some of the loss functions may capture the nature of the outliers better than the other.

When moving from the projected Newton's method to the projected Gauss–Newton method, we may expect a slower convergence, but not necessarily here since the search directions are computed using the truncated PCG iterations, i.e., inexactly. Therefore in the second experiment, we investigate the number of iterations needed for the method to converge. Data are obtained in similar manner as in Table 3 in Kubínová and Nagy [in press] and are shown in Table 5.2. We see that the number of Gauss–Newton iterations is on average higher than for Newton's method with Talwar, which together with no reduce in cost of the computation of the search direction, may outweigh the potential advantage of choice of the loss function.



(a) Satellite single-frame



(b) Carbon ash single-frame

Figure 5.1: Semiconvergence curves – dependence of the relative error of the reconstruction on the size of the regularization parameter λ for various percentages of outliers: standard loss function $t^2/2$ (dashed line) together with the four robust loss functions (solid line).

Table 5.2: Comparison of number of iterations needed for the relative size of the projected gradient to achieve tolerance 10^{-4}. Abbreviation "#it" stands for the number of Newton/Gauss-Newton steps. See Table 3 in Kubínová and Nagy [in press] for comparison.

	# it				
	Newton	Gauss–Newton			
Problem	Talwar	Talwar	Huber	Fair	Logistic
(a) Single-frame					
Satellite	16	15	20	17	19
Satellite random corr. 10%	14	16	25	26	21
Carbon ash	11	11	10	11	9
Carbon ash random corr. 10%	14	14	11	12	12
Satellite added object	15	15	22	26	18
Satellite boundary conditions	25	25	23	17	20
(b) Multi-frame					
Satellite	11	12	14	14	12
Satellite random corr. 10%	13	13	13	15	14
Carbon ash	11	12	12	13	11
Carbon ash random corr. 10%	19	20	11	11	11
Satellite added object	11	12	13	14	12
Satellite boundary conditions	14	23	16	14	15

Bibliography

D. Coleman, P. Holland, N. Kaden, V. Klema, and S. C. Peters. A system of subroutines for iteratively reweighted least squares computations. *ACM Transactions on Mathematical Software (TOMS)*, 6(3):327–336, 1980.

E. Haber. *Computational methods in geophysical electromagnetics*. Mathematics in Industry. SIAM, 2015.

P. C. Hansen, V. Pereyra, and G. Scherer. *Least squares data fitting with applications*. Johns Hopkins University Press, Baltimore, MD, 2013.

M. Kubínová and J. G. Nagy. Robust regression for mixed Poisson–Gaussian model. *Numerical Algorithms*, in press.

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

D. P. O'Leary. Robust regression computation using iteratively reweighted least squares. *SIAM J. Matrix Anal. Appl.*, 11(3):466–480, 1990.

6. Conclusions

Numerical methods for discrete inverse problems represent a very broad field of study. Their common goal is to extract important information from the given (measured) data and suppress the influence of inaccuracies (noise) that are always present. Understanding the properties of noise and its behavior in various numerical methods plays crucial role in designing efficient solution schemes. In this thesis, we focused on iterative methods for linear inverse problems.

In iterative methods based on the Golub–Kahan bidiagonalization, it is possible to track cheaply the propagation of noise from the data to the basis vectors of the corresponding Krylov subspace. The residuals of the methods are linear combinations of these basis vectors. We provided explicit relation between the amount of propagated noise in a particular basis vector and the corresponding coefficient in the linear combination for each of the three considered methods. This enabled us also to assess the regularization properties of CRAIG and compare it to those of LSQR and LSMR without even constructing the residual vectors. For this analysis no a priori information about noise is needed and the results can be extended to computations in finite-precision arithmetic (Chapter 2).

We used the Golub–Kahan bidiagonalization to estimate the noise level in image deblurring problems. We numerically illustrated that the performance of the estimator is reliable as long as the smoothing of the operator is significant with respect to the size of (the high-frequency part of) noise, and that this is independent of the type of noise. When very few measurements are taken, the information about the noise level that the Golub-Kahan bidiagonalization can extract may however be insufficient (Chapter 3).

Next, we dealt with approximation properties of the Krylov subspace methods based on short recurrences in finite-precision arithmetic. We focused on the behavior of residuals of finite-precision Galerkin methods, whose size is known to be prone to severe oscillations, even for problems for which it decays monotonically in exact computations. We showed that, despite these oscillations, after proper aggregation over several iterations, the computed residuals can be linked to the residuals from the exact computation with the same input data. More substantial question in the context of finite-precision Krylov subspace methods remains however the relation between the generated Krylov subspaces and the ideal exact ones, which we studied also via the corresponding Ritz vectors. Modifying some of the known results we established a relation between the computed and the exact Ritz vectors in terms of the convergence of the corresponding Ritz values. There still remains much to be done (Chapter 4).

The last part investigated discrete inverse problems with special combination of mixed Poisson-Gaussian noise and unknown outliers in the data. Combining approaches for the two separate problems (mixed noise and outliers), we derived an objective function with the data-fitting function consisting of inner solution-

dependent weights and an outer robust loss function. We proposed an optimization scheme based on Newton's method and showed that the changing weights limit the choice of the loss functions. This choice can be extended by relaxing Newton's method to a Gauss-Newton method. We modified some of the known stopping criteria to work also in this setting (Chapter 5).

Since discrete inverse problems come from various applications, we will hardly be able to understand all aspects of their solution in the near future. Below we list some open questions directly related to the topics discussed in the thesis:

- Since any method for solving inverse problems has the noisy data as its input, one should be interested in how and where noise propagates during the computation. Are there other iterative methods, for which noise can be tracked cheaply? Can this information be used to derive a stopping criteria or to improve the method?
- To understand the relation between finite-precision and exact Krylov subspace methods, it is essential to understand the relation between the subspaces they generate. Moreover, we need to know, how the solution is determined in these subspaces, for example with respect to the formally prescribed optimality condition, such as norm minimization, which is however not satisfied in finite-precision computations.
- Weighted least squares problems with regularization arising in Chapter 5 can be reformulated as saddle-point problems (p. 103). Performance of standard preconditioners, such as the constraint-style preconditioners, may be very dependent on the ill-conditioning of the (1,1)-block. Deriving a preconditioner robust with respect to this conditioning would represent an important step in numerical solution of weighted least squares problems.

List of publications

Journals

[J2] Kubínová, M., Nagy, J. G. (2018). Robust regression for mixed Poisson–Gaussian model. Numerical Algorithms, in press (published online Jan 2018).

[J1] Hnětynková, I., Kubínová, M., Plešinger, M. (2017). Noise representation in residuals of LSQR, LSMR, and CRAIG regularization. Linear Algebra and its Applications, 533, 357–379.

Peer-reviewed conference proceedings

[P2] Gergelits T., Hnětynková I., Kubínová M. (2018). Relating Computed and Exact Entities in Methods Based on Lanczos Tridiagonalization. In: Kozubek T. et al. (eds) High Performance Computing in Science and Engineering. HPCSE 2017. Lecture Notes in Computer Science, vol 11087 (pp. 73–87). Springer, Cham.

[P1] Hnětynková, I., Kubínová, M., Plešinger, M. (2016). Notes on performance of bidiagonalization-based noise level estimator in image deblurring. In: Handlovičová, A., and Ševčovič, D. (eds) Proceedings of the Conference Algoritmy (pp. 333–342). Publishing House of Slovak University of Technology in Bratislava.

Other relevant publications

[O1] Michenková, M. (2011). Numerical algorithms for low-rank matrix completion problems. Internship report. Swiss Federal Institute of Technology.