Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Ariana N. Brown Date

Inner Product Free Krylov Methods for Inverse Problems

By

Ariana N. Brown
Doctor of Philosophy

Mathematics

James Nagy, Ph.D.
Advisor

Julianne Chung, Ph.D.
Committee Member

Talea Mayo, Ph.D.
Committee Member

Malena Sabaté Landman, Ph.D.
Committee Member

Accepted:

Kimberly Jacob Arriola, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Inner Product Free Krylov Methods for Inverse Problems

By

Ariana N. Brown
B.S., Spelman College, 2020
M.S., Emory University, 2023

Advisor: James Nagy, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Mathematics
2025

Abstract

Inner Product Free Krylov Methods for Inverse Problems

By Ariana N. Brown

Iterative Krylov projection methods have become widely used for solving large-scale linear inverse problems. Certain methods that rely on orthogonality require inner products, which create a bottleneck for parallelization and causes the algorithms to fail in low precision. As a result, there is a need for more effective iterative methods to alleviate this computational burden. This study presents new Krylov projection methods that do not require inner products to solve large-scale linear inverse problems.

The first iterative solver is known as the Changing Minimal Residual Hessenberg method (CMRH). The second is a new extension of CMRH to rectangular systems which we call the least squares LU method (LSLU). We further adapt both approaches to efficiently incorporate Tikhonov regularization. These methods are labeled as Hybrid CMRH and Hybrid LSLU. Each of these techniques are known as quasi-minimal residual methods rather than minimal residual methods. Still, these methods do not offer a way to control how closely the quasi-norm approximates the desired norm. In this work, we also propose a new Krylov method that is both inner product free and minimizes a functional that is theoretically closer to the residual norm. The new scheme combines the conventional CMRH method and the newly proposed LSLU method with a randomized sketch-and-solve technique to solve the strongly overdetermined projected least-squares problem. Extensive numerical examples illustrate the effectiveness of all methods in this dissertation.

Inner Product Free Krylov Methods for Inverse Problems

By

Ariana N. Brown
B.S., Spelman College, 2020
M.S., Emory University, 2023

Advisor: James Nagy, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Mathematics
2025

Acknowledgments

First, I want to thank God for empowering and enabling me to do what would otherwise be impossible. I thank Him for opening the door for me to attend Emory University and granting me favor. To my parents and siblings, thank you for your constant prayers, love, and encouragement. To my church family at ChapelHill Atlanta, thank you for making Atlanta feel like home. I specifically thank Mr. Vernon and Mrs. Felice Messam for their prayers and unwavering support. I also want to express my gratitude to my advisor, Dr. James Nagy, for his guidance and encouragement. Thank you for making the transition from Spelman College to Emory University a positive one. Your consistent support for minority students does not go unnoticed. Lastly, I want to thank each member of my committee. To Dr. Talea Mayo, thank you for your advice and mentorship. Representation matters and your presence reminded me of what was possible. To Dr. Julianne Chung, thank you for providing opportunities for me to share my research with others and teaching me how to see the big picture. To Dr. Malena Sabaté Landman, thank you for helping me find my voice in research. It was a privilege working with everyone. I wish all of my committee members the best in their future endeavors.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Mathematics is found in different areas of study, such as science, medicine, engineering, and even finance. Within these disciplines, we utilize mathematics to formulate the problem and model the solution. The field of inverse problems connects the model and data together by recovering the hidden information from the problem. Inverse problems arise in a variety of applications, including medical and geophysical imaging, machine learning, and image deblurring [12, 23, 43, 44]. In this dissertation, we consider a large linear inverse problem of the form:

$$b = Ax_{\text{true}} + e, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$ models the forward problem, $x_{\text{true}} \in \mathbb{R}^n$ is the unknown solution we want to approximate, $b \in \mathbb{R}^m$ is the vector of observed data, and $e \in \mathbb{R}^m$ represents noise and other measurement errors. The linear system arises from the suitable discretization of linear inverse problems, where the solution may not depend continuously on the data [23]. Given A and b, the goal is to estimate x_{true}, but there are various computational challenges.

In many cases, the number of unknowns n may be very large and the forward model matrix A (and its adjoint) can only be accessed through matrix-vector mul-

tiplications. Thus, iterative methods are often used to compute the approximations of x_{true}. Moreover, these inverse problems belong to a class of ill-posed problems, in the sense that small changes in b can produce large changes in the estimate of x_{true}. This is due to the singular values of A decaying and clustering at zero without any distinguishable gap between consecutive ones. For this reason, regularization must be implemented to diminish the instability invoked from the noise and the ill-posed nature of the problem to recover meaningful approximations of the solution [23].

There are numerous iterative methods to solve inverse problems efficiently. However, the capacity we have to measure and store data is constantly increasing, pushing the limits of traditional least-squares solvers in terms of speed and memory requirements. Recently, different directions have emerged to tackle more challenging scenarios. For example, new methods are being investigated that can use lower precision for storage and computation and benefit from distributed memory implementations. New methods are being developed that can reduce global communication points such as the computation of inner products. In another line of research, randomized numerical linear algebra has emerged as a powerful framework to drastically reduce computational costs. Although initial randomized approaches that were based on obtaining low-rank approximations are not suitable for practical large-scale inverse problems, iterative methods that can exploit randomization are being widely adopted and investigated. This motivates the use of inner product free Krylov methods, which also incorporate randomization techniques to overcome the computational burden.

1.1 Related Work and Contributions

In this dissertation, we introduce new Krylov subspace methods. These methods can be grouped into two categories: Inner product Free Methods and Sketched Inner Product Free Methods.

1.1.1 Inner Product Free Krylov Method

The first technique is known as the Changing Minimal Residual Hessenberg Method (CMRH). CMRH uses the Hessenberg method to construct linearly independent basis vectors. Although the Hessenberg method was mentioned in the literature as early as 1950 (see, e.g.,[15, section 44]) for computing eigenvalues, it was Sadok [38] who introduced the CMRH method in 1999 as a way to solve linear systems. In 2012, Sadok and Szyld [39] investigated the relationship between CMRH and the Generalized Minimum Residual Method (GMRES). This study will investigate the iterative regularization properties of CMRH and introduce a new hybrid variant called the Hybrid Changing Minimal Residual Hessenberg Method.

The second approach is an extension of CMRH to rectangular systems and is referred to as the least squares LU (LSLU) method. It uses a modified Hessenberg iterative algorithm to generate linearly independent bases for Krylov subspaces associated with A^TA and AA^T. This method has a theoretical connection to LU factorization (with partial pivoting) and exhibits iterative regularization properties that are similar to LSQR [34, 41]. The main benefit of LSLU is that no inner products are required during the iterations. This leads to projected quasi-minimum residual problems that are smaller and easier to solve at each iteration. Similar to CMRH, an incorporation of Tikhonov regularization is also explored. This technique is known as Hybrid LSLU.

1.1.2 Sketched Inner Product Free Methods

The remaining methods, sketched CMRH and sketched LSLU, are part of a new family of quasi-minimal residual methods that are closer to the residual norm and are inherently free of inner products. This group is able to preserve the critical information found in the residual norms from the observations in the likelihood function. The proposed approaches combine an inner product free Hessenberg projection for gener-

ating a solution subspace with a randomized sketch-and-solve technique. Extensions for Tikhonov regularization are also examined in this study.

1.2 Overview of Dissertation

The dissertation is organized as follows. In Chapter 2, we present relevant background information. The review covers regularization, Krylov subspace methods, and randomized methods for least squares problems. In Chapter 3, we motivate the use of inner product free Krylov subspace methods. We establish CMRH as an iterative regularization method and include a low-precision arithmetic case study to highlight the pitfalls of inner products in other Krylov subspace methods. We introduce a new inner product free method for solving rectangular large-scale linear inverse problems. We propose two new hybrid methods, Hybrid CMRH (H-CMRH) and Hybrid LSLU (H-LSLU), which are currently the only existing inner product free hybrid methods. In Chapter 4, we combine randomized sketching and inner product free Krylov methods to obtain a more accurate approximation of the objective function to be minimized. Numerical results will illustrate that the proposed algorithm can solve large-scale inverse problems efficiently. We provide closing remarks in Chapter 5. Throughout this study, we assume that $\|\cdot\|$ is the Euclidean norm.

Chapter 2

Background

This chapter provides an overview of relevant background information for this dissertation. In Section 2.1, we review regularization techniques and their ability to combat the perturbations in the solution. We describe Krylov subspace methods in Section 2.2 and emphasize their ability to circumvent computational bottlenecks for large-scale inverse problems. Finally, we motivate the need for randomized sketch-and-solve approaches in Section 2.3.

2.1 Regularization

The field of inverse problems belong to a class of ill-posed problems. An inverse problem is considered ill-posed if it violates at least one of the following requirements [23]:

- Existence: The problem must have a solution.
- Uniqueness: There must be only one solution to the problem.
- Stability: The solution must depend continuously on the data.

The existence and uniqueness condition can be fixed by reformulating the inverse problem. The stability condition is more challenging to "deal with" because a viola-

tion implies that arbitrarily small perturbations of the data can produce arbitrarily large perturbations of the solution [23]. To understand this property, we consider the singular value decomposition of A. We let $A = U\Sigma V^T$, where U and V are orthogonal matrices and Σ is a diagonal matrix with entries satisfying $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$. Without loss of generality, the problem is scaled so that $\sigma_1 = 1$. This implies that:

- The singular values, σ_i, decay to, and cluster at 0, without a significant gap to indicate numerical rank.
- The components $|u_i^T b|$, where u_i is the ith column of U, decay on average faster than the singular values σ_i. This is known as the discrete Picard condition [22].
- The singular vectors v_i (i.e. the columns of V) corresponding to small singular values tend to have more oscillations than the singular vectors corresponding to large singular values.

These properties illustrate the impact the error term e has on the inverse solution:

$$\begin{aligned} x_{\text{inv}} &= A^{-1}b \\ &= A^{-1}(Ax_{\text{true}} + e) \\ &= x_{\text{true}} + A^{-1}e \\ &= x_{\text{true}} + V\Sigma^{-1}U^T e \\ &= x_{\text{true}} + \sum_{i=1}^{n} \frac{u_i^T e}{\sigma_i} v_i. \end{aligned} \tag{2.1}$$

Note that we assume A is invertible.

From (2.1), we see that high frequency components in the error are magnified by the division of small singular values. The computed inverse solution is dominated by these high frequency components, and is in general a very poor approximation of the true solution, x_{true}. Thus, the goal is to reformulate the inverse problem so that the "new problem" is less sensitive to the error term e [23].

2.1.1 The Need for Regularization

The "new" problem must be stabilized or regularized so that the solution is more stable or regular. Regularization techniques are implemented to enforce regularity or smoothness and suppress the unwanted noise components. One class of regularization methods, called filtering, can be formulated as a modification of the inverse solution (1.1). Specifically, a filtered SVD solution is

$$x_{\text{filt}} = \sum_{i=1}^{n} \Phi_i \frac{u_i^T b}{\sigma_i} v_i \tag{2.2}$$

where $\Phi_i \approx 1$ for large σ_i, and $\Phi_i \approx 0$ for small σ_i. That is, the large singular value components of the solution are reconstructed, while the components corresponding to the small singular values are filtered out. Different choices of filtering factors Φ_i lead to different methods; popular choices are truncated SVD, Tikhonov, and Wiener filters [22, 29, 43]. For the purpose of this study we focus on Tikhonov, a type of variational regularization, in conjunction with iterative regularization.

2.1.2 Iterative and Variational Regularization

A common approach for approximating x_{true} is iterative regularization. This consists of applying an iterative solver to the least squares problem

$$\min_{x \in \mathbb{R}^n} \|b - Ax\|, \tag{2.3}$$

where regularization is achieved through early termination of the iterations [10]. The stopping iteration acts as a regularization parameter, which is critical in constructing a solution that is not highly oscillatory or overly smooth.

Alternatively, one can use a variational approach such as Tikhonov regularization,

where the aim is, for example, to solve an optimization problem of the form

$$\min_{x \in \mathbb{R}^n} \|b - Ax\|^2 + \lambda^2 \|x\|^2, \tag{2.4}$$

where λ is the regularization parameter and $\|x\|^2$ is a regularization term [10]. Similar to the role of the stopping iteration in iterative regularization, it is important to choose a good value for λ; if it is chosen too large, the regularized solution is overly smooth and a choice of λ that is too small produces a highly oscillatory solution. Moreover, multiple linear solves might be needed to refine the choice of λ, drastically increasing the computational cost. We also note that the solution of (2.4) can be written in filtered form (2.2), where the filter factors are $\Phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}$. However, computing the SVD for large scale problems is very expensive. Thus, when dealing with large-scale problems, it may be necessary to solve (2.4) using an iterative method.

2.1.3 Hybrid Regularization

Hybrid regularization is a particular combination of variational and iterative regularization. This approach consists of iteratively projecting (2.3) onto a small subspace of increasing dimension and applying variational (e.g., Tikhonov) regularization to the small projected problem, where SVD filtering can be used. In this framework, we only require matrix-vector products with A and possibly A^T, which allows us to avoid explicitly constructing or storing A. It also creates a natural environment, namely, the small projected problems, for estimating a good regularization parameter.

Currently, all existing hybrid regularization algorithms require inner products. In some cases, this can be a computational burden. For example in distributed memory implementations with a large number of processors, the inner products (requiring global communication) can be a limiting factor for efficiency; see, e.g. [2, 35]. Moreover, inner products can also affect the performance of algorithms in low and/or mixed

precision arithmetic, where norms of large vectors can lead to under- or overflow, and where the notion of numerical orthogonality depends on the working floating point arithmetic, which can lead to early stopping of the traditional methods. A few inner product free iterative regularization methods exist but generally exhibit very slow convergence properties, or require information that might be difficult to accurately estimate for large-scale problems. The most popular family of algorithms of this kind consists of Chebyshev semi-iterative methods (see e.g. [2, 5, 18]), which require spectral knowledge about the system matrix A. Alternatively, two simpler algorithms that do not require inner products are Landweber [26, Chapter 6.1.1] and Richardson (first-order) [5, Chapter 7.2.3] methods. Note that the first-order Richardson method corresponds to applying Landweber to the normal equations. Moreover, this is equivalent to gradient descent with a fixed step length that depends on spectral bounds for A, requires positive definiteness of A^TA to converge, and are generally slow to converge.

2.2 Krylov Subspace Methods

Krylov iterative methods are a class of very powerful projection methods that make use of Krylov subspaces. For example, if $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite and $b \in \mathbb{R}^n$ are given, then we can consider the Krylov subspace of the form:

$$\mathcal{K}_k(A, b) = span\{b, Ab, A^2b, ..., A^{k-1}b\}.$$

Optimality conditions are imposed to determine approximate solutions, x_k, in these subspaces. As a result, we can efficiently approximate solutions to large-scale linear inverse problems by projecting the problem onto a smaller subspace with increasing dimension. In this section, we review three Krylov subspace methods that will be used for comparison purposes in this dissertation.

2.2.1 GMRES

The Generalized Minimum Residual (GMRES) Method [36] is a projection technique that iteratively approximates the solution to (2.3), where $m = n$, in a Krylov subspace, $\mathcal{K}_k(A, b)$, of increasing dimension. This iterative method uses the Arnoldi algorithm to compute orthornormal basis vectors of $\mathcal{K}_k(A, b)$. Moreover, Arnoldi produces a $(k+1) \times k$ Hessenberg matrix $H^A_{k+1,k}$ and forms the following relationship:

$$AV_k = V_{k+1} H^A_{k+1,k},$$

where the columns of V_k span $\mathcal{K}_k$. At each iteration, GMRES computes

$$y_k = \arg\min_{y \in \mathbb{R}^k} \| \|b\| e_1 - H^A_{k+1,k} y \|$$

and projects back onto the original subspace using $x_k = V_k y_k$ to approximate the solution. Note that $x_k = \arg\min_{x \in \mathcal{R}(V_k)} \|b - Ax\|$, where $\mathcal{R}(\cdot)$ is used to denote the range of the given operator and e_1 denotes the 1st column of the identity matrix of appropriate size. In others words, the approximate solution is such that the norm of the residual $r_k = b - Ax_k$ is the minimum over all such vectors [39]. An implementation of GMRES can be found in Algorithm 1.

Algorithm 1: GMRES

Require: A, b, x_0, maxiter

Define $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$; $v_1 = r_0/\beta$
for $k = 1, \ldots,$ maxiter **do**
  $w_k = Av_k$
  for $j = 1, \ldots, k$ **do**
    $h(j,k) = (w_k, v_j)$; $w_k = w_k - h(j,k)v_j$
  end for
  $h(k+1,k) = \|w_k\|_2$; If $h(k+1,k) = 0$ set maxiter := k and go to 9;
  $v_{k+1} = w_k / h(k+1,k)$
end for
Define the $(k+1) \times k$ Hessenberg matrix $\overline{H}_k = \{h_{jk}\}_{1 \le j \le k+1, 1 \le k \le maxiter}$
Compute y_k the minimizer of $\|\beta e_1 - \overline{H}_k y\|_2$ and $x_k = x_0 + V_k y_k$

2.2.2 QMR

The Quasi-Minimal Residual (QMR) Algorithm [36] is based on Lanczos bi-orthogonalization, which is an extension to nonsymmetric matrices of the symmetric Lanczos algorithm. This technique builds a pair of bi-orthogonal bases for two subspaces

$$\mathcal{K}_k(A, b) = span\{b, Ab, ..., A^{k-1}b\}$$

and

$$\mathcal{K}_k(A^T, b) = span\{b, A^Tb, ..., (A^T)^{k-1}b\},$$

where $A \in \mathbb{R}^{n\times n}$. QMR, detailed in [36], forms this relation

$$AZ_k = Z_{k+1}\overline{T}_k, \tag{2.5}$$

where $\overline{T}_k$ is a $(k+1) \times k$ tridiagonal matrix

$$\overline{T_k} = \begin{pmatrix} T_k \\ \delta_{k+1}e_k^T \end{pmatrix}$$

and δ_{k+1} is a scalar that satisfies $\delta_{k+1}\beta_{k+1} = (z_{k+1}, w_{k+1})$ (see, e.g., [36, Section 7.1]). Note that the columns of Z_k are not orthonormal. Therefore, at each iteration we are minimizing

$$y_k = \arg\min_{y\in\mathbb{R}^k} \|\|b\|e_1 - \overline{T}_k y\|.$$

The solution, y_k, is used to project back onto the original subspace using the following: $x_k = Z_k y_k$. Note that

$$x_k = \arg\min_{x\in x_0+\mathcal{R}(Z_k)} \|Z_{k+1}^{\dagger}(b - Ax)\|. \tag{2.6}$$

It is also worth noting that because the columns of Z_{k+1} are not orthonormal,

$$\|Z_{k+1}^{\dagger}(b - Ax)\| \neq \|b - Ax\|.$$

This means that (2.6) amounts to minimizing a quasi-residual or semi-norm. An implementation of QMR is presented in Algorithm 7.4 of [36].

2.2.3 LSQR

LSQR [41] uses Golub-Kahan bidiagonalization (GKB) to solve (2.3). GKB, denoted in Algorithm 2, computes two orthonormal bases for the Krylov subspaces $\mathcal{K}_k(A^TA, A^Tb)$ and $\mathcal{K}_k(AA^T, b)$. Similar to other Krylov subspace methods, the GKB algorithm forms the following relations:

$$AW_k = Z_{k+1}\overline{B}_k,$$

$$A^TZ_{k+1} = W_k\overline{B}_k + \mu_{k+1}w_{k+1}e_{k+1}^T,$$

where $W_k = [w_1, ..., w_k] \in \mathbb{R}^{m\times k}$, $Z_k = [z_1, ..., z_k] \in \mathbb{R}^{m\times k}$, and

$$\overline{B_k} = \begin{bmatrix} \mu_1 & & & \\ \nu_2 & \mu_2 & & \\ & \ddots & \ddots & \\ & & \nu_k & \mu_k \\ & & & \nu_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k}.$$

At each iteration, LSQR computes

$$y_k = \arg\min_{y\in\mathbb{R}^k} \|\|b\|e_1 - \overline{B}_k y\|$$

and projects back onto the original subspace using $x_k = W_k y_k$ to approximate the solution [17]. Note that $x_k = \arg \min_{x \in \mathcal{R}(W_k)} \|b - Ax\|$.

Algorithm 2: Golub-Kahn bidiagonalization

Require: A, b

Initialize: $\nu_1 = \|b\|$, $z_1 = b/\nu_1$
Initialize: $w = A^T z_1$, $\mu_1 = \|w\|$, $w_1 = w/\mu_1$
for $j = 2, \ldots, k+1$ **do**
    Compute $z = Aw_{j-1} - \mu_{j-1} z_{j-1}$.
    Set $\nu_j = \|z\|$.
    Take $z_j = z/\nu_j$.
    Compute $w = A^T z_j - \nu_j w_{j-1}$.
    Set $\mu_j = \|w\|$.
    Take $w_j = w/\mu_j$.
end for

2.3 Randomized Methods

We conclude this chapter with a discussion on randomized methods to efficiently solve strongly overdetermined least square problems. We discuss a popular technique to reduce the dimensionality of the problem and present information regarding the best formation of the sketch matrix.

Randomized numerical linear algebra, particularly those approaches involving sketching, has gained increasing popularity, see e.g. [32]. Sketching is a linear dimensionality reduction technique, and there are different ways in which sketching has been used to solve least squares problems. One of the conceptually simplest approaches is to sketch-and-solve, which can be used to find approximate solutions of least-squares problems where the system matrix is tall and skinny. This was originally proposed in [40] and has gained a lot of attention due its simplicity and probabilistic guarantees. In particular, we can define a sketching matrix $S \in \mathbb{R}^{s \times m}$, such that the following subspace embedding property is satisfied for any vector $a \in \mathbb{R}^m$ in a given

set of vectors

$$(1-\epsilon)\|a\| \leq \|Sa\| \leq (1+\epsilon)\|a\|. \tag{2.7}$$

For this to be a dimensionality reduction technique, we typically assume that $s \ll m$. Even if this is a very favorable property, it is not trivial to construct such matrices deterministically in practice, in the sense that (2.7) is guaranteed for any $a \in \mathbb{R}^m$. However, the analytical properties of random matrices can be used to construct sketching matrices $S \in \mathbb{R}^{s\times m}$ that will satisfy (2.7) with high probability. Note that this is a special case of a random subspace embedding; for a formal definition, see e.g. [32, Chapter 8.1].

Moreover, there exist different choices of random matrices in the literature that can be computationally cheap to construct and apply. The easiest class to analyze and implement is that of Gaussian embeddings, where each entry of $S \in \mathbb{R}^{s\times m}$ is an independent draw of a Gaussian distribution with zero mean and variance $1/s$. Note that applying a Gaussian sketch to a vector has an $O(sm)$ cost (the explicit storage of the sketch matrix is also $O(sm)$), see [32, Chapter 8.3]. When dealing with high dimensional problems one can also use structured random embeddings, which can reduce the storage and application costs. The most well-used sketches in this case are the subsampled randomized trigonometric transforms (SRTT), subsampled random Fourier transform (SRFT), and the sparse sign embeddings. However, the latter sketches require sufficiently efficient implementations to be faster than a simple Gaussian sketch in practice. For simplicity, we will only use Gaussian embeddings, but all results can be generalized to the use of other sketching techniques.

One of the most popular uses of sketching is to find approximate solutions of least-squares problems. Suppose we have a tall and skinny matrix $A \in \mathbb{R}^{m\times n}$, where $m \gg n$, then we can define a sketching matrix $S \in \mathbb{R}^{s\times m}$, such that $s \ll m$ and is assumed to be a small multiple of n. A representation of the sketched matrix $SA \in \mathbb{R}^{s\times n}$, of smaller dimension than A, can be observed in Figure 2.1. The sketch-

and-solve method for finding an approximate solution to the least-squares problem (2.3) involves solving the following minimization problem

$$\min_{x \in \mathbb{R}^n} \|S(Ax - b)\|, \tag{2.8}$$

where S is a sketch matrix, see, e.g. [32, Chapter 10.3], [40]. Note that, when using Gaussian sketching, the solution of (2.8) is an unbiased estimator of the solution of the original least-squares problem (2.3) provided that the matrix A has full column rank.

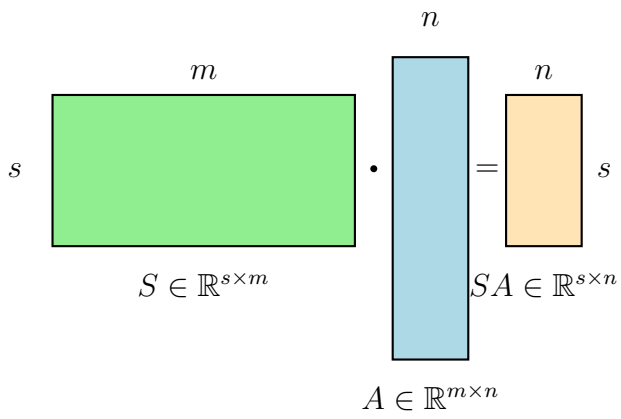

Figure 2.1: Schematic representation of the sketching of a matrix A using a sketch S.

The fact that A needs to be tall and skinny seems to be a restrictive property, since a lot of applications do not give rise to systems where the matrices are naturally in this form. However, it has been proposed to use randomized techniques in combination with Krylov methods, where the projections give rise to such tall and skinny matrices [20]. Specifically, for inverse problems, we know that a good approximation of the solution can be found in a Krylov subspace of small dimension involving the right-hand side b, A, and possibly A^T. Thus, as we will describe in Chapter 3, one can construct a basis for the relevant Krylov subspace(s) and use sketching to solve the projected least-sqaures problems, since this now involves a tall and skinny matrix even if A is not.

Chapter 3

Inner Product Free Krylov Subspaces Methods

This chapter, which is based on the collaborative works [6, 7], focuses on inner product free Krylov methods, where the first step is to build nonorthogonal bases for the relevant Krylov subspaces. In Section 3.1 we describe CMRH and establish this method as an iterative regularization method. In Section 3.2 we create an inner product free quasi-minimum residual method for rectangular systems. We conclude this chapter with a hybrid extension of CMRH and LSLU in Section 3.3 and numerical results in Section 3.4.

3.1 The Changing Minimal Residual Hessenberg Method

The Changing Minimal Residual Hessenberg Method is an algorithm that iteratively approximates the solution to (2.3) in a Krylov subspace of increasing dimension,

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

where $A \in \mathbb{R}^{n\times n}$, $r_0 = b - Ax_0$, and x_0 is the initial guess of the solution. We define the Krylov matrix as

$$\tilde{V}_k = [r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0] \in \mathbb{R}^{n\times k}, \tag{3.1}$$

where $\tilde{V}_{k+1} = [r_0, A\tilde{V}_k]$. Since $\tilde{V}_k$ is an ill-conditioned matrix for even small k, it is not explicitly constructed but only used to motivate a scheme to compute a linearly independent basis for $\mathcal{K}_k(A, r_0)$ called the Hessenberg process [15].

For the Hessenberg process, consider the LU factorization,

$$\tilde{V}_k = \tilde{L}_k\tilde{U}_k, \tag{3.2}$$

where $\tilde{L}_k \in \mathbb{R}^{n\times k}$ is a unit lower triangular matrix and $\tilde{U}_k \in \mathbb{R}^{k\times k}$ is an upper triangular matrix. The algorithm recursively computes the columns of $\tilde{L}_k$ but does not explicitly compute the LU factorization of $\tilde{V}_k$. From the construction of $\tilde{V}_{k+1}$, we can write the following relation,

$$\tilde{V}_{k+1}\begin{bmatrix} 0_{1\times k} \\ I_k \end{bmatrix} = \tilde{L}_{k+1}\tilde{U}_{k+1}\begin{bmatrix} 0_{1\times k} \\ I_k \end{bmatrix} = A\tilde{V}_k = A\tilde{L}_k\tilde{U}_k, \tag{3.3}$$

where $0_{1\times k}$ is a row vector of zeros with dimensions $1 \times k$. Following [39], we define an upper Hessenberg matrix as

$$\tilde{H}_{k+1,k} = \tilde{U}_{k+1}\begin{bmatrix} 0_{1\times k} \\ I_k \end{bmatrix}\tilde{U}_k^{-1} \in \mathbb{R}^{(k+1)\times k}, \tag{3.4}$$

where $k < n$. Furthermore, combining (3.3) and (3.4), we get the Hessenberg relation:

$$A\tilde{L}_k = \tilde{L}_{k+1}\tilde{H}_{k+1,k}, \tag{3.5}$$

where the columns of $\tilde{L}_k$ form a linearly independent basis for $\mathcal{K}_k(A, r_0)$. Algorithm 3 contains a description of the Hessenberg Process for square A [27].

Algorithm 3: Hessenberg Process for Square A

Require: A, b, x_0, maxiter

$r_0 = b - Ax_0$, $\beta = e_1^T r_0$; $\tilde{l}_1 = r_0/\beta$
for $k = 1, \ldots,$ maxiter **do**
  $u = A\tilde{l}_k$
  for $j = 1, \ldots, k$ **do**
    $\tilde{H}(j,k) = u(j)$; $u = u - \tilde{H}(j,k)\tilde{l}_j$
  end for
  $\tilde{H}(k+1,k) = u(k+1)$; $\tilde{l}_{k+1} = u/\tilde{H}(k+1,k)$
end for

Notice that at each iteration of the Hessenberg process, we require one matrix-vector multiplication with A and no computations of inner products. From Algorithm 3, we can observe that the Hessenberg process will break down if $\beta = e_1^T r_0 = 0$ or $\tilde{H}(k+1,k) = 0$. To avoid this and to avoid severe ill-conditioning in the basis vectors, i.e, the columns of $\tilde{L}_k$, [27] introduces the Hessenberg process with pivoting, which is provided in Algorithm 4.

Building off the Hessenberg process, CMRH is an iterative projection algorithm for computing an approximate solution to (2.3), where at each iteration k, the following least-squares problem is solved

$$\min_{x \in \mathcal{R}(\tilde{L}_k)} \|\tilde{L}_{k+1}^{\dagger}(b - Ax)\|, \tag{3.6}$$

where $\tilde{L}_{k+1}^{\dagger}$ is the pseudoinverse of $\tilde{L}_{k+1}$ and $\mathcal{R}(\cdot)$ is used to denote the range of the given operator. With initial guess x_0, $r_0 = b - Ax_0$, and using (3.5), we can see that

the solution is given by $x_k = x_0 + \tilde{L}_k y_k$ where

$$\begin{aligned} \|\tilde{L}_{k+1}^{\dagger}(b - A(x_0 + \tilde{L}_k y))\| &= \|\tilde{L}_{k+1}^{\dagger}(r_0 - A\tilde{L}_k y)\| \\ &= \|\tilde{L}_{k+1}^{\dagger}(r_0 - \tilde{L}_{k+1}\tilde{H}_{k+1,k} y)\| \\ &= \|\beta e_1 - \tilde{H}_{k+1,k} y\| \end{aligned}$$

and

$$y_k = \arg\min_{y \in \mathbb{R}^k} \|\beta e_1 - \tilde{H}_{k+1,k} y\|. \tag{3.7}$$

Here, β is either the first element of r_0 if using Algorithm 3 or the element of r_0 with the largest absolute value if using Algorithm 4.

Algorithm 4: Hessenberg Process with Pivoting

Require: A, b, x_0, maxiter
Define $p = [1, 2, \ldots, n]^T$, and let $r_0 = b - Ax_0$
Determine i such that $|r_0(i)| = \|r_0\|_\infty$
$\beta = r_0(i)$; $\tilde{l}_1 = r_0/\beta$; $p(1) \Leftrightarrow p(i)$
for $k = 1, \ldots,$ maxiter **do**
  $u = A\tilde{l}_k$
  for $j = 1, \ldots, k$ **do**
    $\tilde{H}(j,k) = u(p(j))$; $u = u - \tilde{H}(j,k)\tilde{l}_j$
  end for
  if $k < n$ and $u \neq 0$ **then**
    Determine $i \in \{k+1, \ldots, n\}$ such that $|u(p(i))| = \|u(p(k+1:n))\|_\infty$
    $\tilde{H}(k+1,k) = u(p(i))$; $\tilde{l}_{k+1} = u/\tilde{H}(k+1,k)$; $p(k+1) \Leftrightarrow p(i)$
  else
    $\tilde{H}(k+1,k) = 0$; Stop
  end if
end for

3.1.1 The relationship between CMRH and GMRES

In this section, we recall the theoretical relationship between the basis vectors produced by GMRES and CMRH, as well as bounds on the difference between their relative residual norms. We follow closely the work of Szyld and Sadok [39] and in-

clude additional details on the derivations for completeness. This will guide the proof for the hybrid version of the algorithm in Section 3.3. This does not, however, provide a sufficient explanation regarding the regularization properties of CMRH, which is explained in Section 3.1.2.

Similar to CMRH, one can derive the GMRES method by considering the QR factorization of the Krylov matrix defined in (3.1):

$$\tilde{V}_k = Q_k \tilde{R}_k \tag{3.8}$$

where $Q_k \in \mathbb{R}^{n\times k}$ has orthonormal columns and $\tilde{R}_k$ is an upper triangular matrix. This algorithm does not explicitly compute the QR factorization, but instead recursively computes the columns of Q_k. From [39] we consider the following relationship:

$$\tilde{V}_{k+1} \begin{bmatrix} 0_{1\times k} \\ I_k \end{bmatrix} = Q_{k+1}\tilde{R}_{k+1} \begin{bmatrix} 0_{1\times k} \\ I_k \end{bmatrix} = A\tilde{V}_k = AQ_k\tilde{R}_k, \tag{3.9}$$

where $0_{1\times k}$ is a row vector of zeros with dimensions $1 \times k$. Since $\tilde{R}_{k+1}$, $\tilde{R}_k$, and $\tilde{R}_k^{-1}$ are upper triangular matrices, we observe from (3.9) that

$$H^A_{k+1,k} = Q^T_{k+1}AQ_k = \tilde{R}_{k+1} \begin{bmatrix} 0_{1\times k} \\ I_k \end{bmatrix} \tilde{R}_k^{-1} \in \mathbb{R}^{(k+1)\times k}, \tag{3.10}$$

is an upper Hessenberg matrix where $k < n$. Note that the columns of Q_k form an orthonormal basis of $\mathcal{K}_k(A, r_0)$. Moreover, combining (3.9) and (3.10) creates the Arnoldi relation

$$AQ_k = Q_{k+1}H^A_{k+1}. \tag{3.11}$$

The derivation of (3.11) and (3.5) naturally provides a mapping between the Arnoldi and Hessenberg bases. Consider the LU and QR factorizations of the Krylov

matrix $\tilde{V}_k$ in (3.2) and (3.8), respectively; then we can define

$$\tilde{V}_k = \tilde{L}_k \tilde{U}_k = Q_k \tilde{R}_k. \tag{3.12}$$

Now, let $R_k = \tilde{R}_k \tilde{U}_k^{-1}$: this is a $k \times k$ upper triangular matrix. Using (3.12), the lower triangular matrix $\tilde{L}_k$ can be written as a composition of an orthogonal matrix and an upper right triangular matrix,

$$\tilde{L}_k = Q_k \tilde{R}_k \tilde{U}_k^{-1}.$$

This corresponds to a QR factorization of $\tilde{L}_k$,

$$\tilde{L}_k = Q_k R_k. \tag{3.13}$$

In addition to (3.13), we can rewrite the Arnoldi relation in (3.11) as

$$A\tilde{L}_k R_k^{-1} = \tilde{L}_{k+1} R_{k+1}^{-1} H_{k+1,k}^A. \tag{3.14}$$

Comparing the Hessenberg (3.5) and Arnoldi (3.14) relations, and following [39], we provide the following proposition.

Proposition 1. *Let $\tilde{H}_{k+1,k}$ and $H_{k+1,k}^A$ be the Hessenberg matrices associated to the Hessenberg and Arnoldi processes, respectively, at iteration k, then*

$$\tilde{H}_{k+1,k} = R_{k+1}^{-1} H_{k+1,k}^A R_k,$$

or, equivalently, $H_{k+1,k}^A = R_{k+1} \tilde{H}_{k+1,k} R_k^{-1}$.

Proposition 1 is used to establish residual bounds for CMRH in the following sense: the residual norm associated to the approximate solution provided by CMRH

at each iteration is close to the residual norm associated to the solution provided by GMRES if the condition number of R_{k+1} does not grow too quickly. This can be observed in the following theorem, originally proved in [39]. Here, we restate it and provide additional details of the proof in order to set the stage for an analogous reasoning of Hybrid CMRH in Section 3.3.

Theorem 1. *Let r_k^G and r_k^C be the GMRES and CMRH residuals at the kth iteration beginning with the same initial residual r_0, respectively. Then*

$$\|r_k^G\| \leq \|r_k^C\| \leq \kappa(R_{k+1})\|r_k^G\| \tag{3.15}$$

where $\kappa(R_{k+1}) = \|R_{k+1}\|\|R_{k+1}^{-1}\|$ is the condition number of R_{k+1}.

Proof. First, we prove the left inequality in (3.15). Consider the residual as a function of the solution:

$$r(x) = b - Ax.$$

Then, the kth residual norm associated to the approximated solution produced by GMRES is:

$$\|r_k^G\| = \|b - Ax_k^G\| = \min_{x \in \mathcal{K}_k} \|r(x)\|$$

Since x_k^G and x_k^C are in the Krylov subspace $\mathcal{K}_k$, then by definition:

$$\min_{x \in \mathcal{K}_k} \|r(x)\| \leq \|r_k^C\| = \|r(x_k^C)\|.$$

Hence, $\|r_k^G\| \leq \|r_k^C\|$.

Now we prove the right inequality in (3.15). Since r_k^C and r_k^G are in the $\mathcal{K}_{k+1}(A, r_0)$ subspace, we can write r_k^C and r_k^G as a linear combination of any basis of $\mathcal{K}_{k+1}(A, r_0)$. Using the Hessenberg relation, the decomposition of the Krylov matrix is:

$$\tilde{V}_{k+1} = \tilde{L}_{k+1}\tilde{U}_{k+1}.$$

This implies range$(\tilde{L}_{k+1})$ = range$(\tilde{V}_{k+1}) = \mathcal{K}_{k+1}(A, r_0)$. Therefore, and using (3.13), there exist u_k^C and w_k^C in $\mathbb{R}^{k+1}$ such that

$$r_k^C = \tilde{L}_{k+1} u_k^C = Q_{k+1} R_{k+1} u_k^C = Q_{k+1} w_k^C$$

with $R_{k+1} u_k^C = w_k^C$. Analogously, there exist u_k^G and w_k^G in $\mathbb{R}^{k+1}$ such that

$$r_k^G = \tilde{L}_{k+1} u_k^G = Q_{k+1} R_{k+1} u_k^G = Q_{k+1} w_k^G \tag{3.16}$$

with $R_{k+1} u_k^G = w_k^G$.

Consider the optimality conditions of CMRH. As stated above, $r_k^C = \tilde{L}_{k+1} u_k^C$. This implies that $\tilde{L}_{k+1}^{\dagger} r_k^C = u_k^C$. Hence, $\|\tilde{L}_{k+1}^{\dagger} r_k^C\| = \|u_k^C\|$ so

$$\|u_k^C\| = \min_{x \in \mathcal{K}_k} \|\tilde{L}_{k+1}^{\dagger}(b - Ax)\| = \min_{x \in \mathcal{K}_k} \|\tilde{L}_{k+1}^{\dagger} r(x)\|. \tag{3.17}$$

Using (3.17) and the fact that x_k^G is in $\mathcal{K}_k(A, r_0)$ then $\|u_k^C\| \leq \|u_k^G\|$. Thus

$$\|u_k^C\| \leq \|u_k^G\| = \|R_{k+1}^{-1} w_k^G\| \leq \|R_{k+1}^{-1}\| \|w_k^G\| = \|R_{k+1}^{-1}\| \|r_k^G\|,$$

where the equalities in the above relation come from (3.16). On the other hand, by employing (3.16) we have:

$$\|r_k^C\| = \|\tilde{L}_{k+1} u_k^C\| \leq \|\tilde{L}_{k+1}\| \|u_k^C\|.$$

Putting the above inequalities together gives the following relation:

$$\begin{aligned} \|r_k^C\| &= \|\tilde{L}_{k+1} u_k^C\| \\ &\leq \|\tilde{L}_{k+1}\| \|u_k^C\| \\ &\leq \|\tilde{L}_{k+1}\| \|R_{k+1}^{-1}\| \|r_k^G\|. \end{aligned}$$

Recall that $\tilde{L}_{k+1}$ has a QR decomposition (3.13) of the form $\tilde{L}_{k+1} = Q_{k+1}R_{k+1}$, where Q_{k+1} has orthonormal columns. Therefore, $\|\tilde{L}_{k+1}\| = \|Q_{k+1}R_{k+1}\| = \|R_{k+1}\|$. This results in the following:

$$\|r_k^C\| \leq \|\tilde{L}_{k+1}\|\|R_{k+1}^{-1}\|\|r_k^G\| = \|R_{k+1}\|\|R_{k+1}^{-1}\|\|r_k^G\| = \kappa(R_{k+1})\|r_k^G\|.$$

Thus, we conclude that $\|r_k^G\| \leq \|r_k^C\| \leq \kappa(R_{k+1})\|r_k^G\|$. □

If we compare the computational cost between GMRES and CMRH, each algorithm requires one matrix-vector product per iteration, which is typically the most expensive computation in an iterative method. Additional costs related to vector operations are also similar, except that GMRES requires $k+1$ inner products at the kth iteration, while no inner products need to be computed in CMRH. Note that, in cases where the number of required iterations is large and matrix-vector products can be done very efficiently, the growing number of inner products required by GMRES can be a computational bottleneck. In these cases, CMRH presents a computational advantage with respect to GMRES.

3.1.2 Regularizing properties of CMRH

This section is devoted to providing a theoretical understanding that underlies the regularizing mechanisms of CMRH. These theoretical results are supported by technical numerical examples, which are designed to highlight the behavior of CMRH in this context rather than its performance. Note that a variety of additional numerical examples on the performance of CMRH for ill-posed problems can be found in Section 3.4.

As an iterative solver, CMRH produces a sequence of approximate solutions. In this study we observe that, in early iterations, these approximations begin to converge toward the true solution, x_{true}; however, if the iterations are not stopped early, the

approximate solutions eventually deviate from x_{true} due to the ill-conditioning of A and the noise, e (see Section 2.1). This is known as semiconvergence [23]. To overcome semiconvergence, we must stop CMRH at the right time.

An analysis of the SVD of the projected matrix $\tilde{H}_{k+1,k}$ from (3.5) provides a basis for understanding the regularizing properties of CMRH. Moreover, our hypothesis is that the approximated solutions produced by CMRH mirror regularized (filtered) solutions, so the spectral filtering properties of CMRH are also studied empirically in this section.

3.1.2.1 Singular Value Decomposition Analysis

It has been observed that the singular values of the projected matrix $H^A_{k+1,k}$ in (3.11) for GMRES (and the corresponding projected matrix for QMR [17]) tend to approximate the large singular components of A. This helps to explain the regularizing properties of these Krylov methods and the initial decay of the relative residual norms of QMR and GMRES [17].

In this section, we compare the largest singular values of the matrix A with the singular values of the upper Hessenberg matrix $\tilde{H}_{k+1,k}$ defined in (3.5) and obtained at each iteration of CMRH. To analyze how well the singular values from $\tilde{H}_{k+1,k}$ approximate those of A, in Figures 3.1 and 3.2 we plot the largest singular values of the full matrix A using horizontal lines. The dots represent the singular values of the projected matrices at each iteration k (see top plots of Figures 3.1 and 3.2). As another way of visualizing this, the bottom plots display the singular values of the upper Hessenberg matrix at different iterations of CMRH against the singular values of A. Each figure represents a different 1D inverse problem; the first three (Deriv2, Heat, and Shaw) are from the Regularization Tools Package [25], and the fourth example (Spectra) is a 1D signal restoration problem, with a matrix modeling

a Gaussian blur. Specifically, the entries of A are given by

$$a_{ij} = \frac{1}{\varsigma\sqrt{2\pi}} \exp\left(-\frac{(i-j)^2}{2\varsigma^2}\right), \tag{3.18}$$

with $\varsigma = 2$, and x_{true} is a simulated x-ray spectrum [42].

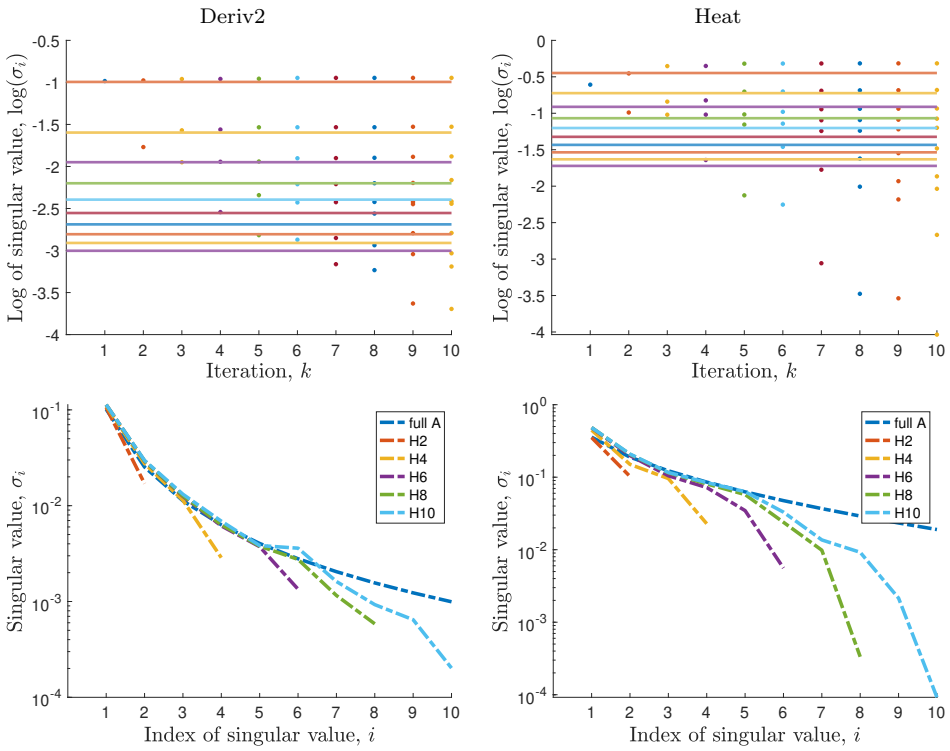

Figure 3.1: Singular values for Deriv2-example 2 and Heat (see [25] for Deriv2 details).

From Figures 3.1 and 3.2, it seems that the quality of the singular value approximations depends on the conditioning of A. To investigate this, we construct the following experiment: Consider the matrix $A \in \mathbb{R}^{n\times n}$ and the vector x_{true} generated from Spectra test problem (3.18). We compute the SVD of A and keep the singular vectors from the orthogonal matrices U and V. We replace the diagonal matrix containing the singular values of A with a diagonal matrix of S whose diagonal elements are defined as:

$$S_{kk}^{(i)} = e^{c_i * k} \quad \text{for} \quad k = 1, ..., n,$$

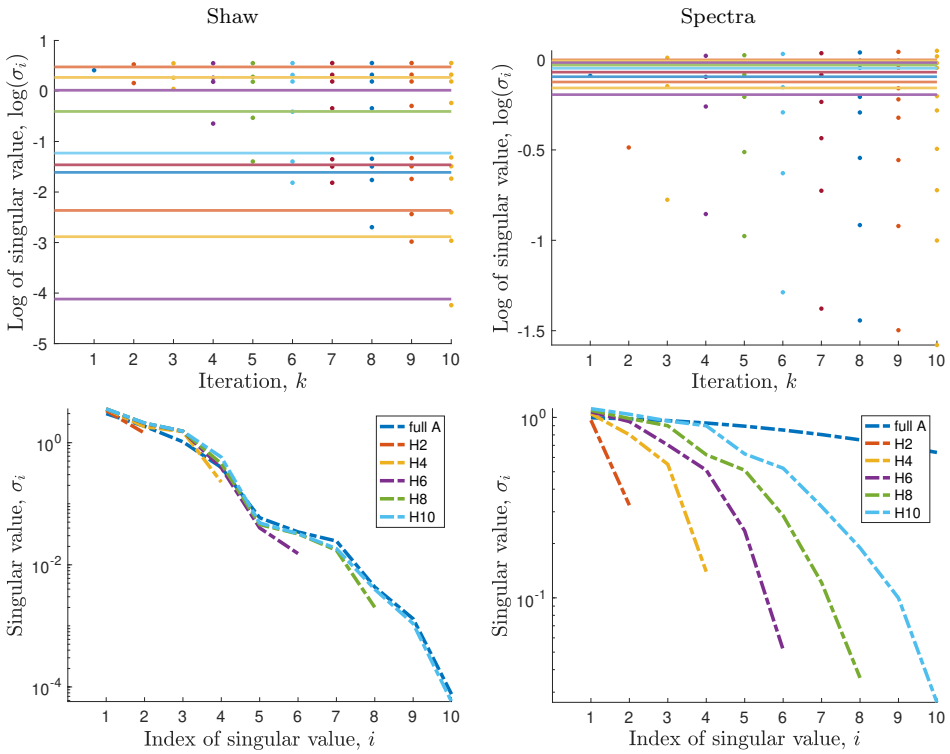

Figure 3.2: Singular values for Shaw and Spectra.

where c_i are entries from the vector $c = [-2, -1, -0.5, -0.25]$. For these different matrices $S^{(i)}$, we compare the largest singular values of $A^{(i)} = US^{(i)}V^T$ with those associated with the Hessenberg matrices obtained using the Hessenberg process associated to $A^{(i)}$ and the right-hand side $b^{(i)} = A^{(i)}x_{\text{true}}$. The results are displayed in Figures 3.3 and 3.4. Indeed, it appears that the more ill-conditioned A is, the better the singular values of the Hessenberg matrices associated to the Hessenberg process approximate the largest singular values of A. This result is coherent with what we observed from the previous experiments, for example, in the test problems Shaw for Figure 3.2 and Deriv2 for Figure 3.1.

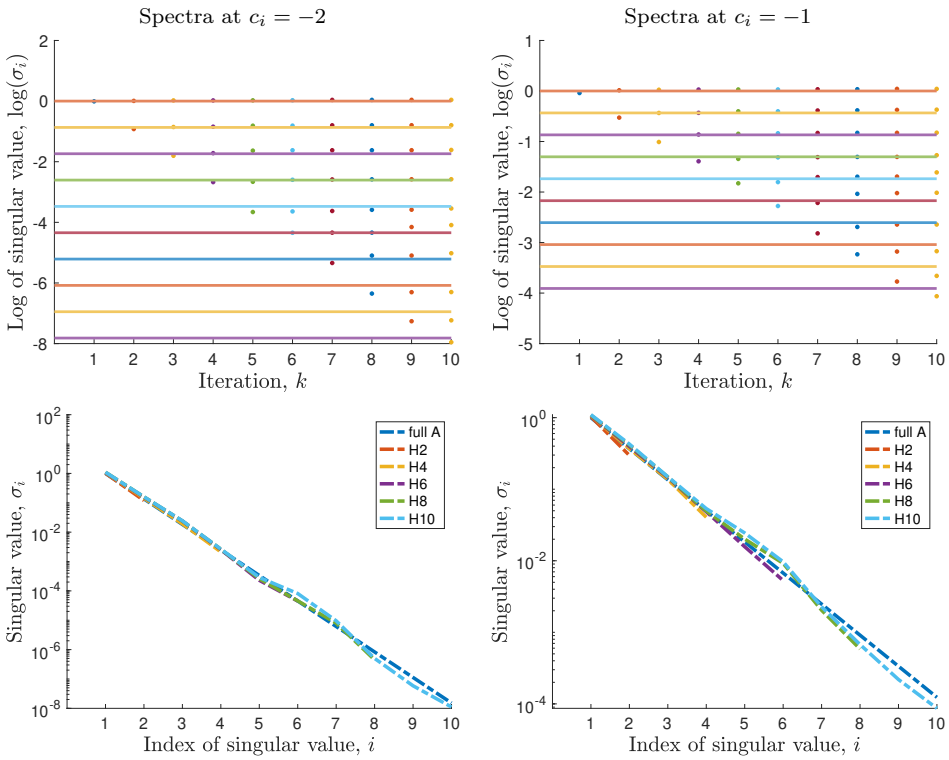

Figure 3.3: Singular values for modified Spectra.

3.1.2.2 Spectral Filtering Properties

From Section 2.1, we find that an approximate solution to (1.1) is said to be regularized using spectral filtering if it can be written in the form

$$x = \sum_{i=1}^{n} \Phi_i \frac{u_i^T b}{\sigma_i} v_i, \tag{3.19}$$

where u_i and v_i correspond to left and right singular vectors of A, and σ_i are its singular values in nonincreasing order of magnitude. Spectral filtering methods are a wide class of methods, including, for example, Tikhonov regularization; see, e.g.,[23, Chapter 4] and Section 2.1.

A natural question arises when studying the regularization properties of CMRH: Is there an empirical relationship between the approximate solutions computed with CMRH and a solution with spectral filtering of the form in (3.19)? Using (3.19),

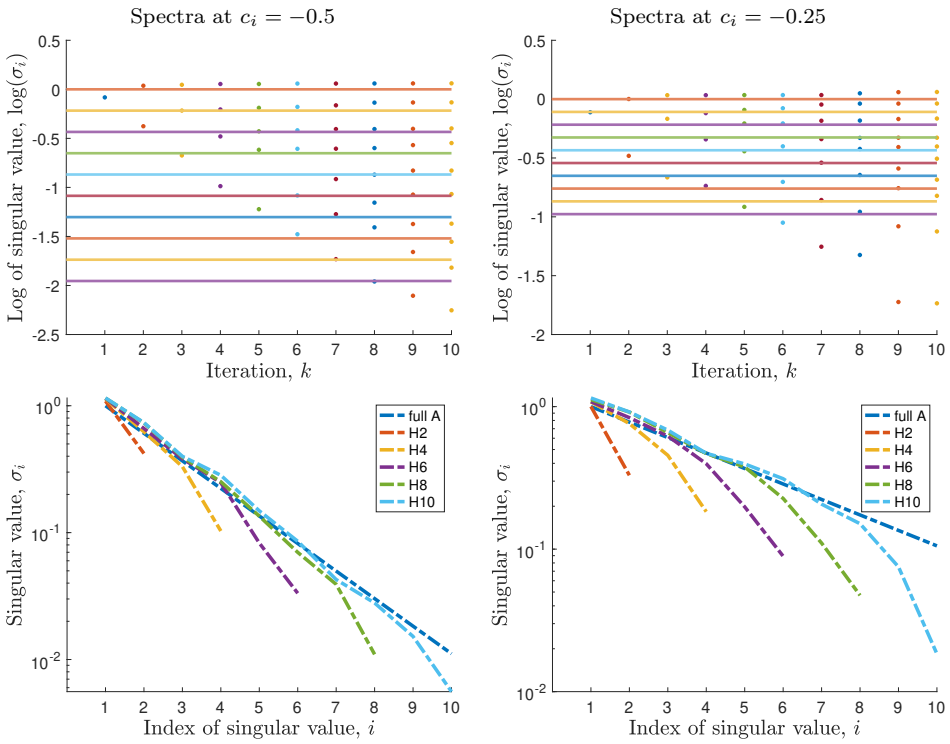

Figure 3.4: Singular values for modified Spectra.

one can compute the empirical associated filter factors Φ_i to a given approximated solution at the kth iteration, x_k, as

$$\Phi_i = \frac{v_i^T x_k}{u_i^T b}\sigma_i, \tag{3.20}$$

assuming the SVD of A is available.

Consider the 1D signal restoration problem Spectra solved using CMRH and GMRES. In Figure 3.5, we observe the relative error norm histories for three different noise levels. For each noise level, the plots in Figure 3.6 display the empirical filter factors for some iterations of CMRH and GMRES. These are indicated with a marker in Figure 3.5 and aim to illustrate three iterations that are common for all noise levels and the last iteration before the relative error increases.

First, one can observe that for all combinations of noise levels and iterations displayed in the plots in Figure 3.6, the empirical filter factors for CMRH and GMRES

have a very similar behavior. This is also reminiscent of the filter factors corresponding to the TSVD, which present a sharp phase transition between filter factors being 1 and 0. Also note that this phase transition moves to the right (that is, happens at a higher value of k) as the iterations increase; in other words, fewer iterations correspond to more filtering. This a very positive result, as it indicates that CMRH is an effective method for filtering highly oscillatory noisy components if the iterations are stopped early.

We also observe that the different noise levels (in this small regime) do not affect significantly the filter factors in the first iterations. We reiterate that in order for CMRH to mimic spectral filtering regularization, the iterative solver must terminate before the relative error increases.

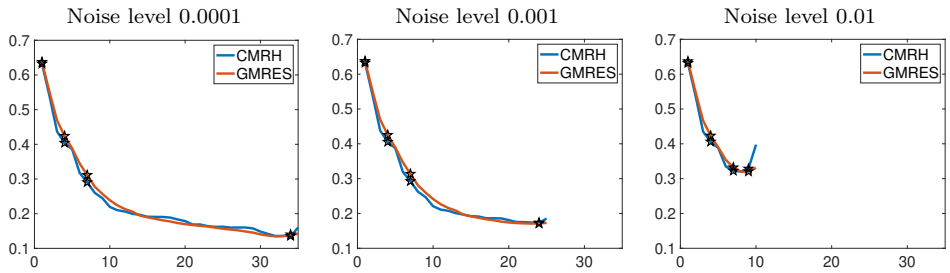

Figure 3.5: Relative error norms for the approximate solutions to test problem Spectra computed using CMRH and GMRES for different noise levels. The empirical filter factors for both methods at the indicated markers can be observed in the plots in Figure 3.6.

3.2 Least Squares with LU Factorization

Now, we introduce an extension of CMRH, known as LSLU, to problems where A is a rectangular matrix. This is a common scenario in the field of inverse problems. We show that the Hessenberg process for rectangular matrices is directly related to applying the Hessenberg process to the normal equations. In a similar fashion to CMRH, we impose a quasi-minimal residual optimality condition. This is comparable

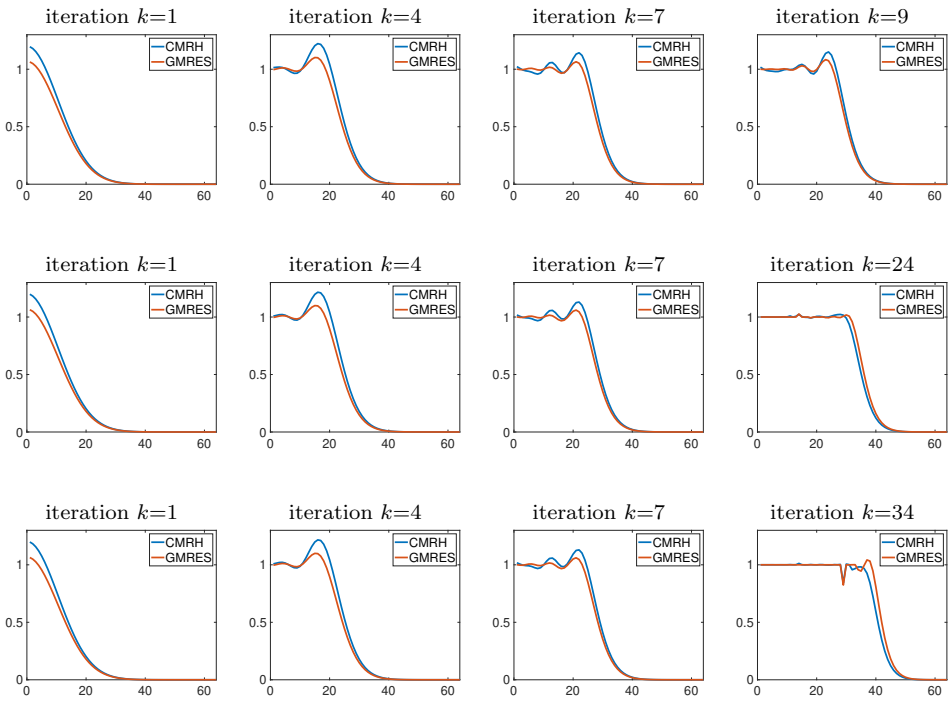

Figure 3.6: Empirical filter factors Φ_i for solutions of test problem Spectra computed using GMRES and CMRH at iterations corresponding to the markers in Figure 3.5. The top row corresponds to left plot in Figure 3.5 (noise level 0.001), the middle row corresponds to the middle plot of Figure 3.5 (noise level 0.001), and the bottom row corresponds to the right plot in Figure 3.5 (noise level 0.01).

to the process found in LSQR [41] where the basis vectors are built using Golub-Kahan bidiagonalization (or symmetric Lanczos on the normal equations) and the optimality conditions that are imposed minimize the residual norm.

3.2.1 Extension of the Hessenberg Process

In this section, we describe an extension of the Hessenberg process for rectangular systems with $A \in \mathbb{R}^{m\times n}$, where the main difference is that we require two sets of basis vectors, one for each of the following Krylov subspaces,

$$\mathcal{K}_k(A^TA, v_0) = span\{v_0, A^TAv_0, (A^TA)^2v_0, \ldots, (A^TA)^{k-1}v_0\}, \tag{3.21}$$

$$\mathcal{K}_k(AA^T, r_0) = span\{r_0, AA^Tr_0, (AA^T)^2r_0, \ldots, (AA^T)^{k-1}r_0\}, \tag{3.22}$$

where $r_0 = b - Ax_0$ and $v_0 = A^T r_0$, with x_0 as an initial guess of the solution. We introduce an iterative method called LSLU that minimizes an oblique projection of the residual and exploits components of the Hessenberg process for efficient computation.

Assume that no breakdowns occur in the initialization process. The Hessenberg method for rectangular systems, detailed in Algorithm 5, generates at the kth iteration vectors l_{k+1} and d_{k+1} such that

$$AL_k = D_{k+1}H_{k+1,k} \tag{3.23}$$

$$A^T D_{k+1} = L_{k+1}W_{k+1}, \tag{3.24}$$

where $L_k \in \mathbb{R}^{n\times k}$ is unit lower triangular, $D_k \in \mathbb{R}^{m\times k}$ is unit lower triangular, $H_{k+1,k} \in \mathbb{R}^{(k+1)\times k}$ is upper Hessenberg, and $W_{k+1} \in \mathbb{R}^{(k+1)\times(k+1)}$ is upper triangular.

From (3.23) and (3.24), we obtain the following Hessenberg relationships:

$$A^T AL_k = A^T D_{k+1}H_{k+1,k} = L_{k+1}W_{k+1}H_{k+1,k}, \tag{3.25}$$

$$AA^T D_{k+1} = AL_{k+1}W_{k+1} = D_{k+2}H_{k+2,k+1}W_{k+1}, \tag{3.26}$$

where the products $W_{k+1}H_{k+1,k}$ and $H_{k+2,k+1}W_{k+1}$ are upper Hessenberg matrices. Comparing (3.25) with (3.5), we see that the Hessenberg process for rectangular systems is equivalent to the Hessenberg process for square systems applied to the normal equations, with system matrix $A^T A$ and resulting upper Hessenberg matrix $W_{k+1}H_{k+1,k}$.

Algorithm 5: Hessenberg Process for Rectangular Systems

Require: A, b, x_0, maxiter

Define $r_0 = b - Ax_0$, $\beta = e_1^T r_0$; $d_1 = r_0/\beta$
for $k = 1, \ldots,$ maxiter **do**
  $q = A^T d_k$
  for $j = 1, \ldots, k-1$ **do**
    $W(j,k) = q(j)$; $q = q - W(j,k) l_j$
  end for
  $W(k,k) = q(k)$; $l_k = q/W(k,k)$
  $u = A l_k$
  for $j = 1, \ldots, k$ **do**
    $H(j,k) = u(j)$; $u = u - H(j,k) d_j$
  end for
  $H(k+1,k) = u(k+1)$; $d_{k+1} = u/H(k+1,k)$;
end for

Similar to the derivation in Section 3.1.1, we can define Krylov matrices,

$$P_k = [v_0, A^T A v_0, (A^T A)^2 v_0, \ldots, (A^T A)^{k-1} v_0] \in \mathbb{R}^{n \times k}, \tag{3.27}$$

$$C_k = [r_0, A A^T r_0, (A A^T)^2 r_0, \ldots, (A A^T)^{k-1} r_0] \in \mathbb{R}^{m \times k}, \tag{3.28}$$

whose columns span (3.21) and (3.22) respectively. It follows that $P_{k+1} = [v_0, A^T A P_k]$ and $C_{k+1} = [r_0, A A^T C_k]$.

Note that, by construction, the columns of P_k and L_k span the same space for all k. In particular, the vector p_j can be written as a linear combination of the columns of L_j, which correspond to the first j columns of the matrix L_k, for all $j \leq k$. This means that there exists an upper triangular matrix U_k such that $P_k = L_k U_k$, and since L_k is unit lower triangular, this corresponds to an LU factorization of P_k. Note that Algorithm 5 does not explicitly compute this LU factorization, but recursively generates the columns of L_k. Applying this factorization provides the

following relation:

$$P_{k+1}\begin{bmatrix}0_{1\times k}\\ I_k\end{bmatrix} = L_{k+1}U_{k+1}\begin{bmatrix}0_{1\times k}\\ I_k\end{bmatrix} = A^TAP_k = A^TAL_kU_k, \tag{3.29}$$

where $0_{1\times k}$ is a row vector of zeros with dimensions $1 \times k$.

From (3.25), we find that $A^TAL_k = L_{k+1}W_{k+1}H_{k+1,k}$. (3.29) can be written as:

$$L_{k+1}U_{k+1}\begin{bmatrix}0_{1\times k}\\ I_k\end{bmatrix} = A^TAL_kU_k. \tag{3.30}$$

Since U_{k+1}, U_k, and U_k^{-1} are upper triangular matrices, we observe from (3.30) that

$$L_{k+1}U_{k+1}\begin{bmatrix}0_{1\times k}\\ I_k\end{bmatrix}U_k^{-1} = A^TAL_k. \tag{3.31}$$

Thus, by (3.25) and (3.31) we recover the upper Hessenberg matrix:

$$W_{k+1}H_{k+1,k} = U_{k+1}\begin{bmatrix}0_{1\times k}\\ I_k\end{bmatrix}U_k^{-1}, \tag{3.32}$$

where $k < n$.

Following an analogous argument to the one used for L_k, the columns of C_k and D_k span the same space for all k by construction, and there exists an upper triangular matrix G_{k+1} such that $C_{k+1} = D_{k+1}G_{k+1}$ corresponds to an LU factorization of C_{k+1}. Applying this factorization to $C_{k+1} = [r_0, AA^TC_k]$ provides the following relation:

$$C_{k+2}\begin{bmatrix}0_{1\times(k+1)}\\ I_{k+1}\end{bmatrix} = D_{k+2}G_{k+2}\begin{bmatrix}0_{1\times(k+1)}\\ I_{k+1}\end{bmatrix} = AA^TC_{k+1} = AA^TD_{k+1}G_{k+1}, \tag{3.33}$$

where $0_{1\times(k+1)}$ is a row vector of zeros with dimensions $1 \times (k+1)$.

From (3.26), we find that $AA^T D_{k+1} = D_{k+2} H_{k+2,k+1} W_{k+1}$. (3.33) can be written as:

$$D_{k+2} G_{k+2} \begin{bmatrix} 0_{1\times(k+1)} \\ I_{k+1} \end{bmatrix} = AA^T D_{k+1} G_{k+1}. \tag{3.34}$$

Since G_{k+2}, G_{k+1}, and G_{k+1}^{-1} are upper triangular matrices, we observe from (3.34) that

$$D_{k+2} G_{k+2} \begin{bmatrix} 0_{1\times(k+1)} \\ I_{k+1} \end{bmatrix} G_{k+1}^{-1} = AA^T D_{k+1}. \tag{3.35}$$

Thus, by (3.26) and (3.35) we recover the upper Hessenberg matrix:

$$H_{k+2,k+1} W_{k+1} = G_{k+2} \begin{bmatrix} 0_{1\times k+1} \\ I_{k+1} \end{bmatrix} G_{k+1}^{-1}, \tag{3.36}$$

where $k < m$.

From Algorithm 5, we find that the process will breakdown if either $\beta = 0$, $H(k+1,k) = 0$, or $W(k,k) = 0$. Moreover in floating point arithmetic, additional errors can occur if these values are small and not exactly zero. To avoid division by small numbers, in practice we implement the Hessenberg process with pivoting, which is given in Algorithm 6. Notice that pivoting is required twice.

Algorithm 6: Hessenberg Process with Pivoting for Rectangular Systems

Require: A, b, x_0, maxiter

Define $t = [1, 2, \ldots, m]^T$, $g = [1, \ldots, n]^T$.
$r_0 = b - Ax_0$
Determine i such that $|r_0(i)| = \|r_0\|_\infty$
$\beta = r_0(i)$; $d_1 = r_0/\beta$; $t(1) \Leftrightarrow t(i)$
for $k = 1, \ldots,$ maxiter **do**
  $q = A^T d_k$
  for $j = 1, \ldots, k-1$ **do**
    $W(j,k) = q(g(j))$; $q = q - W(j,k)l_j$
  end for
  if $k < n$ and $q \neq 0$ **then**
    Determine $i \in \{k, \ldots, n\}$ such that $|q(g(i))| = \|q(g(k:n))\|_\infty$
    $W(k,k) = q(g(i))$; $l_k = q/W(k,k)$; $g(k) \Leftrightarrow g(i)$
  else
    break
  end if
  $u = Al_k$
  for $j = 1, \ldots, k$ **do**
    $H(j,k) = u(t(j))$; $u = u - H(j,k)d_j$
  end for
  if $k < m$ and $u \neq 0$ **then**
    Determine $i \in \{k+1, \ldots, m\}$ such that $|u(t(i))| = \|u(t(k+1:m))\|_\infty$
    $H(k+1,k) = u(t(i))$; $d_{k+1} = u/H(k+1,k)$; $t(k+1) \Leftrightarrow t(i)$
  else
    break
  end if
end for

LSLU is a new iterative projection method. At each iteration k, LSLU finds an approximate solution for (2.3) by minimizing the following least squares problem:

$$\min_{x \in x_0 + \mathcal{R}(L_k)} \|D_{k+1}^{\dagger}(b - Ax)\|, \tag{3.37}$$

where $D_{k+1}^{\dagger}$ is the pseudoinverse of D_{k+1}. Note that the functional in (3.37) can be considered as an approximation to the residual norm of the original problem, which is similar to the QMR method; see e.g., [36] and Section 2.2.2. More specifically, considering $x = x_0 + L_k y$ and $r_0 = b - Ax_0$, the objective function in (3.37) can be

written as

$$\begin{aligned} \|D_{k+1}^{\dagger}(b - A(x_0 + L_k y))\| &= \|D_{k+1}^{\dagger}(r_0 - AL_k y)\| \\ &= \|D_{k+1}^{\dagger}(r_0 - D_{k+1}H_{k+1,k}y)\| \\ &= \|\beta e_1 - H_{k+1,k}y\|, \end{aligned}$$

where β is either the first entry of r_0 (Algorithm 5) or the entry of r_0 with the largest absolute value (Algorithm 6). Thus, at iteration k we solve the following subproblem,

$$y_k = \arg\min_{y \in \mathbb{R}^k} \|\beta e_1 - H_{k+1,k}y\|,$$

which is of smaller dimension compared to the original problem. Once y_k is computed, then $x_k = x_0 + L_k y_k$ provides an approximate solution of the original least squares problem (2.3). The algorithm corresponding to this method can be found as a special case of the hybrid method described in Section 3.3.

In Algorithm 6, we must find the entry with the largest absolute value of r_0 and two other vectors at each iteration. For any given vector x, this corresponds to finding i such that $|x(i)| = \|x\|_\infty$, which can be costly as computing $\|x\|_\infty$ requires global communication. In order to avoid this in LSLU, we propose the following pivoting alternative: select a small random sample of entries from r_0, u, q, and choose the largest value (in magnitude) in that sample. Provided that the selection is "large enough", we achieve a reasonable approximate solution.

As an illustration, we use the the PRtomo example from IR Tools [16] (see Section 3.4 for details), and we use $25, 50$, and 100 samples to approximate the infinity norm. Note that the samples are only being used for determining the pivot, and that the number of samples is tiny compared to the more than 65000 elements in each of the vectors. We provide relative reconstruction error norms, $\|x_{\text{true}} - x_k\|/\|x_{\text{true}}\|$, per iteration in Figure 3.7, where the sampled LSLU approach denoted 'LSLU inf est'

performs similar to the LSLU approach where the pivots are determined using the actual infinity norm, denoted 'LSLU'. Note that sometimes 'LSLU inf est' seems to perform better than 'LSLU' in that one can observe a delay in the semi-convergence phenomenon. However, the minimal attained error norm for 'LSLU inf est' is comparable or marginally larger than the one corresponding to the version with standard partial pivoting. The relative reconstruction error norms per iteration of LSQR are provided to illustrate that the new LSLU method is competitive with existing methods. Additional numerical results will be provided in Section 3.4.

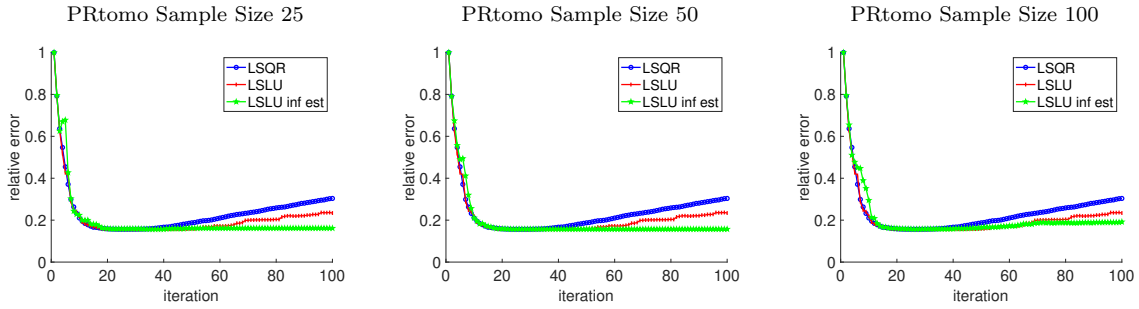

Figure 3.7: Relative reconstruction error norms per iteration for LSLU with pivoting using the infinity norm, compared to using the estimated infinity norm as the maximum from a set of randomly sampled coefficients (denoted 'LSLU inf est'). Results for LSQR are provided for reference.

3.2.2 Theoretical bounds for the residual norm of LSLU

In this section, we derive a bound on the difference between the residual norms of solutions computed using LSLU and LSQR. Let $\hat{R}_{k+1}$ be an upper triangular matrix from the QR decomposition of D_{k+1}. We show that if the condition number of $\hat{R}_{k+1}$ does not grow too quickly, the residual norms associated with the approximate solutions of LSLU and LSQR at each iteration are close to each other. This is critical in understanding the regularizing properties of LSLU.

Theorem 2. *Let r_k^{QR} and r_k^{LU} be the LSQR and LSLU residuals at the kth iteration*

beginning with the same initial guess $x_0 = 0$*, respectively. Then*

$$\|r_k^{QR}\| \leq \|r_k^{LU}\| \leq \kappa(\hat{R}_{k+1})\|r_k^{QR}\| \tag{3.38}$$

where $\kappa(\hat{R}_{k+1}) = \|\hat{R}_{k+1}\|\|\hat{R}_{k+1}^{-1}\|$ *is the condition number of* $\hat{R}_{k+1}$.

Proof. First, we prove the left inequality in (3.38). Consider the residual as a function of the solution,

$$r(x) = b - Ax.$$

Then, the residual norm associated with the approximate solution at the kth iteration of LSQR is

$$\|r_k^{QR}\| = \|b - Ax_k^{QR}\| = \min_{x \in \mathcal{K}_k(A^TA, A^Tb)} \|r(x)\|.$$

Since x_k^{QR} and x_k^{LU} are in the Krylov subspace $\mathcal{K}_k(A^TA, A^Tb)$, then by definition,

$$\min_{x \in \mathcal{K}_k(A^TA, A^Tb)} \|r(x)\| \leq \|r(x_k^{LU})\| = \|r_k^{LU}\|.$$

Hence, $\|r_k^{QR}\| \leq \|r_k^{LU}\|$.

Now we prove the right inequality in (3.38). Since r_k^{QR} and r_k^{LU} are in the subspace $\mathcal{K}_k(AA^T, b)$, we can write r_k^{QR} and r_k^{LU} as a linear combination of any basis of $\mathcal{K}_k(AA^T, b)$. Using the Hessenberg relation, the LU decompositions of P_{k+1} and C_{k+1} are

$$C_{k+1} = D_{k+1}G_{k+1}$$

$$P_{k+1} = L_{k+1}U_{k+1}.$$

This implies that $\mathcal{R}(D_{k+1}) = \mathcal{R}(C_{k+1}) = \mathcal{K}_{k+1}(AA^T, b)$ and $\mathcal{R}(L_{k+1}) = \mathcal{R}(P_{k+1}) = \mathcal{K}_{k+1}(A^TA, A^Tb)$. Therefore, using the QR decomposition of $D_{k+1} = \hat{U}_{k+1}\hat{R}_{k+1}$ and

$L_{k+1} = V_{k+1}\tilde{R}_{k+1}$, there exist z_k^{LU} and s_k^{LU} in $\mathbb{R}^{k+1}$ such that

$$r_k^{LU} = D_{k+1}z_k^{LU} = \hat{U}_{k+1}\hat{R}_{k+1}z_k^{LU} = \hat{U}_{k+1}s_k^{LU} \tag{3.39}$$

with $s_k^{LU} = \hat{R}_{k+1}z_k^{LU}$. Analogously, there exist z_k^{QR} and s_k^{QR} in $\mathbb{R}^{k+1}$ such that

$$r_k^{QR} = D_{k+1}z_k^{QR} = \hat{U}_{k+1}\hat{R}_{k+1}z_k^{QR} = \hat{U}_{k+1}s_k^{QR} \tag{3.40}$$

with $s_k^{QR} = \hat{R}_{k+1}z_k^{QR}$.

Consider the optimality conditions of LSLU. As stated above, $r_k^{LU} = D_{k+1}z_k^{LU}$. This implies that $D_{k+1}^{\dagger}r_k^{LU} = z_k^{LU}$. Hence, $\|D_{k+1}^{\dagger}r_k^{LU}\| = \|z_k^{LU}\|$ so

$$\|z_k^{LU}\| = \min_{x\in\mathcal{R}(L_k)} \|D_{k+1}^{\dagger}(b - Ax)\| = \min_{x\in\mathcal{R}(L_k)} \|D_{k+1}^{\dagger}r(x)\|. \tag{3.41}$$

Using (3.41) and the fact that x_k^{QR} is in $\mathcal{K}_k(A^TA, A^Tb)$ then

$$\|z_k^{LU}\| = \min_{x\in\mathcal{K}_k(A^TA,A^Tb)} \|D_{k+1}^{\dagger}r(x)\| \leq \|D_{k+1}^{\dagger}r(x_k^{QR})\| = \|z_k^{QR}\|.$$

Thus

$$\|z_k^{LU}\| \leq \|z_k^{QR}\| = \|\hat{R}_{k+1}^{-1}s_k^{QR}\| \leq \|\hat{R}_{k+1}^{-1}\|\|s_k^{QR}\| = \|\hat{R}_{k+1}^{-1}\|\|r_k^{QR}\|,$$

where the equalities in the above relation come from (3.40). On the other hand, applying (3.39) gives:

$$\|r_k^{LU}\| = \|D_{k+1}z_k^{LU}\| \leq \|D_{k+1}\|\|z_k^{LU}\|.$$

Putting the above inequalities together gives the following relation:

$$\begin{aligned} \|r_k^{LU}\| &= \|D_{k+1} z_k^{LU}\| \\ &\leq \|D_{k+1}\| \|z_k^{LU}\| \\ &\leq \|D_{k+1}\| \|\hat{R}_{k+1}^{-1}\| \|r_k^{QR}\|. \end{aligned}$$

Recall that D_{k+1} has a QR decomposition of the form $D_{k+1} = \hat{U}_{k+1}\hat{R}_{k+1}$, where $\hat{U}_{k+1}$ is an orthogonal matrix. Therefore, $\|D_{k+1}\| = \|\hat{U}_{k+1}\hat{R}_{k+1}\| = \|\hat{R}_{k+1}\|$. This results in the following:

$$\|r_k^{LU}\| \leq \|D_{k+1}\| \|\hat{R}_{k+1}^{-1}\| \|r_k^{QR}\| = \|\hat{R}_{k+1}\| \|\hat{R}_{k+1}^{-1}\| \|r_k^{QR}\| = \kappa(\hat{R}_{k+1}) \|r_k^{QR}\|.$$

Thus, we conclude that $\|r_k^{QR}\| \leq \|r_k^{LU}\| \leq \kappa(\hat{R}_{k+1}) \|r_k^{QR}\|$. □

3.3 Hybrid CMRH and Hybrid LSLU

In this section, we consider the hybrid variants of CMRH and LSLU for solving large-scale linear inverse problems. In addition to being inner product free, these methods can compute regularized solutions efficiently and with automatically selected regularization parameters. In Section 3.3.1 we provide theoretical bounds for the residual norms of Hybrid CMRH and Hybrid LSLU and conclude with addressing some computational considerations in Section 3.3.2.

Consider the standard Tikhonov regularization problem (2.4). Hybrid CMRH and Hybrid LSLU are iterative methods where the solution at the kth iteration is computed as the solution to the optimization problem,

$$\min_{x \in x_0 + \mathcal{R}(L_k)} \|M_{k+1}^{\dagger}(b - Ax)\|^2 + \lambda_k^2 \|N_k^{\dagger} x\|^2, \tag{3.42}$$

where $M_{k+1}^{\dagger} = \tilde{L}_{k+1}^{\dagger}$ and $N_k^{\dagger} = \tilde{L}_k^{\dagger}$ for Hybrid CMRH. Likewise, $M_{k+1}^{\dagger} = D_{k+1}^{\dagger}$ and $N_k^{\dagger} = L_k^{\dagger}$ for Hybrid LSLU. Similar to the non-hybrid cases, the residual norm is replaced by a semi-norm and the regularization term also includes a semi-norm. It can be shown that solving (3.42) is equivalent to solving

$$y_{\lambda,k} = \arg\min_{y \in \mathbb{R}^k} \|\beta e_1 - Z_{k+1,k} y\|^2 + \lambda_k^2 \|y\|^2, \tag{3.43}$$

where $Z = \tilde{H}_{k+1,k}$ for Hybrid CMRH, $Z = H_{k+1,k}$ for Hybrid LSLU, and β is the largest entry in r_0 (when considering the Hessenberg method implementation with pivoting). We project back onto the original subspace using $x_k = x_0 + N_k y_{\lambda,k}$. Note that $N_k = L_k$ for Hybrid LSLU and $N_k = \tilde{L}_k$ for Hybrid CMRH. An implementation of Hybrid CMRH and Hybrid LSLU with pivoting is provided in Algorithm 7 and Algorithm 8. These algorithms corresponds to CMRH and LSLU when $\lambda = 0$. As a hybrid approach, the regularization parameter, denoted as λ_k in (3.42) and (3.43), can be selected at each iteration. Various regularization parameter selection techniques will be discussed in Section 3.3.2.1.

Algorithm 7: Hybrid CMRH

Compute $r_0 = b - Ax_0$
$p = [1, 2, ..., n]^T$
Determine i_0 such that $|r_0(i_0)| = \|r_0\|_\infty$
$\beta = r_0(i_0)$; $\tilde{l}_1 = r_0/\beta$; $p_1 \Leftrightarrow p_{i_0}$
for $k = 1, ..., m$ **do**
    $u = A\tilde{l}_k$
    for $j = 1, ..., k$ **do**
        $\tilde{h}(j,k) = u(p(j))$; $u = u - \tilde{h}(j,k)\tilde{l}_j$
    end for
    if $k < n$ and $u \neq 0$ **then**
        Determine $i_0 \in \{k+1, ..., n\}$ such that $|u(p(i_0))| = \|u(p(k+1:n))\|_\infty$
        $\tilde{h}(k+1,k) = u(p(i_0))$; $\tilde{l}_{k+1} = u/\tilde{h}(k+1,k)$; $p_{k+1} \Leftrightarrow p_{i_0}$
    else
        $\tilde{h}(k+1,k) = 0$; Stop.
    end if
end for
Define the $(k+1) \times k$ Hessenberg matrix
Implement regularization parameter scheme (λ_k): optimal or GCV
Compute the minimizer of $\|\beta e_1 - \tilde{H}_{k+1,k} y\|^2 + \lambda_k^2 \|y\|^2$ and $x_k = x_0 + \tilde{L}_k y_{\lambda,k}$

Algorithm 8: Hybrid LSLU

Require: A, b, x_0, maxiter, RegParam

Define $t = [1, 2, \ldots, m]^T$, $g = [1, \ldots, n]^T$.
$r_0 = b - Ax_0$
Determine i such that $|r_0(i)| = \|r_0\|_\infty$
$\beta = r_0(i)$; $d_1 = r_0/\beta$; $t(1) \Leftrightarrow t(i)$
for $k = 1, \ldots,$ maxiter **do**
  $q = A^T d_k$
  for $j = 1, \ldots, k-1$ **do**
    $W(j,k) = q(g(j))$; $q = q - W(j,k)l_j$
  end for
  if $k < n$ and $q \neq 0$ **then**
    Determine $i \in \{k, \ldots, n\}$ such that $|q(g(i))| = \|q(g(k:n))\|_\infty$
    $W(k,k) = q(g(i))$; $l_k = q/W(k,k)$; $g(k) \Leftrightarrow g(i)$
  else
    break
  end if
  $u = Al_k$
  for $j = 1, \ldots, k$ **do**
    $H(j,k) = u(t(j))$; $u = u - H(j,k)d_j$
  end for
  if $k < m$ and $u \neq 0$ **then**
    Determine $i \in \{k+1, \ldots, m\}$ such that $|u(t(i))| = \|u(t(k+1:m))\|_\infty$
    $H(k+1,k) = u(t(i))$; $d_{k+1} = u/H(k+1,k)$; $t(k+1) \Leftrightarrow t(i)$
  else
    break
  end if
  Find regularization parameter λ_k according to the RegParam scheme (using e.g. the methods discussed in section 3.3.2).
  Compute $y_{\lambda_k,k}$ as the minimizer of $\|\beta e_1 - H_{k+1,k}y\|_2^2 + \lambda_k^2\|y\|_2^2$
  $x_k = x_0 + L_k y_{\lambda_k,k}$
end for

3.3.1 Theoretical bounds for the residual norms of Hybrid CMRH and Hybrid LSLU

The residual norms of Hybrid CMRH and Hybrid LSLU can be bounded in a similar fashion to its non-hybrid version. In Section 3.3.1.1 we discuss the difference between the residual norm of Hybrid CMRH and Hybrid GMRES. The residual norms of Hybrid LSQR and Hybrid LSLU will be investigated in Section 3.3.1.2.

3.3.1.1 Residual Norm of Hybrid CMRH

To understand the regularization properties of Hybrid CMRH, we investigate the residual bounds of Hybrid CMRH and Hybrid GMRES. Please note that $A \in \mathbb{R}^{n\times n}$. Let

$$\overline{L}_{k+1} = \begin{bmatrix} \tilde{L}_{k+1} & 0 \\ 0 & \tilde{L}_k \end{bmatrix}, \tag{3.44}$$

with $\tilde{L}_k, \tilde{L}_{k+1}$ defined by the Hessenberg relation (3.5). We assume λ to be fixed and find that if the condition number of $\overline{L}_{k+1}$ does not grow too quickly, then the residual norm associated to the solution obtained with Hybrid CMRH is close to the residual norm of the solution obtained with Hybrid GMRES.

Theorem 3. *Let hr_k^G and hr_k^C be the Hybrid GMRES and Hybrid CMRH residuals at the kth iteration beginning with the same initial residual r_0, respectively. Then*

$$\|hr_k^G\| \leq \|hr_k^C\| \leq \kappa(\overline{L}_{k+1})\|hr_k^G\|, \tag{3.45}$$

where $\kappa(\overline{L}_{k+1}) = \|\overline{L}_{k+1}\|\|\overline{L}_{k+1}^{\dagger}\|$ is the condition number of $\overline{L}_{k+1}$.

Proof. First, we prove the left inequality in (3.45). We can define the hybrid residual as a function of the solution:

$$hr(x) = \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} A \\ \lambda I \end{bmatrix} x.$$

Since x_k^G and x_k^C are in the Krylov subspace $\mathcal{K}_k$, by the optimality conditions of Hybrid GMRES:

$$\|hr_k^G\| = \min_{x\in\mathcal{K}_k} \|hr(x)\| \leq \|hr(x_k^C)\| = \|hr_k^C\|.$$

Hence $\|hr_k^G\| \leq \|hr_k^C\|$.

Now we prove the right inequality in (3.45). Since for any $x \in \mathcal{K}_k(A, r_0)$ we have that $Ax - b \in \mathcal{K}_{k+1}(A, r_0)$=range($\tilde{L}_{k+1}$); and $x_k^G, x_k^C \in \mathcal{K}_k(A, r_0)$= range($\tilde{L}_k$), we can write hr_k^C and hr_k^G as a linear combination of the columns of $\overline{L}_{k+1}$ defined in (3.44). Let $hr_k^C = \overline{L}_{k+1} u_k^C$ and $hr_k^G = \overline{L}_{k+1} u_k^G$. This implies that $u_k^C = \overline{L}_{k+1}^\dagger hr_k^C$ and $u_k^G = \overline{L}_{k+1}^\dagger hr_k^G$. Hence, $\|\overline{L}_{k+1}^\dagger hr_k^C\| = \|u_k^C\|$ and $\|\overline{L}_{k+1}^\dagger hr_k^G\| = \|u_k^G\|$. By the optimality conditions of Hybrid CMRH:

$$\|u_k^C\| = \min_{x \in \mathcal{K}_k} \left\| \begin{bmatrix} L_{k+1}^\dagger & 0 \\ 0 & L_k^\dagger \end{bmatrix} \left(\begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} A \\ \lambda I \end{bmatrix} x \right) \right\| = \min_{x \in \mathcal{K}_k} \|\overline{L}_{k+1}^\dagger hr(x)\|. \tag{3.46}$$

Using (3.46) and the fact that x_k^G is in $\mathcal{K}_k(A, r_0)$ then $\|u_k^C\| \leq \|u_k^G\|$. Thus

$$\|u_k^C\| \leq \|u_k^G\| = \|\overline{L}_{k+1}^\dagger hr_k^G\| \leq \|\overline{L}_{k+1}^\dagger\| \|hr_k^G\|.$$

On the other hand,

$$\|hr_k^C\| = \|\overline{L}_{k+1} u_k^C\| \leq \|\overline{L}_{k+1}\| \|u_k^C\|.$$

Putting the above inequalities together gives the following relation:

$$\begin{aligned} \|hr_k^C\| &= \|\overline{L}_{k+1} u_k^C\| \\ &\leq \|\overline{L}_{k+1}\| \|u_k^C\| \\ &\leq \|\overline{L}_{k+1}\| \|\overline{L}_{k+1}^\dagger\| \|hr_k^G\| \\ &= \kappa(\overline{L}_{k+1}) \|hr_k^G\|, \end{aligned}$$

so we conclude that $\|hr_k^G\| \leq \|hr_k^C\| \leq \kappa(\overline{L}_{k+1}) \|hr_k^G\|$. □

3.3.1.2 Residual Norm of Hybrid LSLU

Now, we will analyze the theoretical bounds of Hybrid LSLU and Hybrid LSQR to provide insight on the regularizing properties of Hybrid LSLU. Note that $A \in \mathbb{R}^{m \times n}$.

Let λ be fixed and

$$\overline{D}_{k+1} = \begin{bmatrix} D_{k+1} & 0 \\ 0 & L_k \end{bmatrix}, \tag{3.47}$$

with D_{k+1} and L_k defined by the Hessenberg relations (3.23) and (3.24). We find that if the condition number of $\overline{D}_{k+1}$ does not grow too quickly, then the residual norm associated to the solution obtained with Hybrid LSLU is close to the residual norm of the solution obtained with Hybrid LSQR.

Theorem 4. *Let hr_k^{QR} and hr_k^{LU} be the Hybrid LSQR and Hybrid LSLU residuals at the kth iteration beginning with the same initial residual r_0, respectively. Then*

$$\|hr_k^{QR}\| \leq \|hr_k^{LU}\| \leq \kappa(\overline{D}_{k+1})\|hr_k^{QR}\| \tag{3.48}$$

where $\kappa(\overline{D}_{k+1}) = \|\overline{D}_{k+1}\|\|\overline{D}_{k+1}^{\dagger}\|$ is the condition number of $\overline{D}_{k+1}$.

Proof. First, we prove the left inequality in (3.48). We can define the hybrid residual as a function of the solution,

$$hr(x) = \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} A \\ \lambda I \end{bmatrix} x.$$

Since x_k^{QR} and x_k^{LU} are in the Krylov subspace $\mathcal{K}_k(A^TA, A^Tb)$, by the optimality conditions of Hybrid LSQR,

$$\|hr_k^{QR}\| = \min_{x \in \mathcal{R}(L_k)} \|hr(x)\| \leq \|hr(x_k^{LU})\| = \|hr_k^{LU}\|.$$

Hence $\|hr_k^{QR}\| \leq \|hr_k^{LU}\|$.

Now we prove the right inequality in (3.48). Since $b - Ax \in \mathcal{K}_{k+1}(AA^T, b) = \mathcal{R}(D_{k+1})$, then for any $x \in \mathcal{K}_k(A^TA, A^Tb)$ and x_k^{QR}, $x_k^{LU} \in \mathcal{K}_k(A^TA, A^Tb) = \mathcal{R}(L_k)$ we can write hr_k^{LU} and hr_k^{QR} as a linear combination of the columns of $\overline{D}_{k+1}$ defined

in (3.47).

Let $hr_k^{LU} = \overline{D}_{k+1} z_k^{LU}$ and $hr_k^{QR} = \overline{D}_{k+1} z_k^{QR}$. This implies that $z_k^{LU} = \overline{D}_{k+1}^{\dagger} hr_k^{LU}$ and $z_k^{QR} = \overline{D}_{k+1}^{\dagger} hr_k^{QR}$. Hence, $\|\overline{D}_{k+1}^{\dagger} hr_k^{LU}\| = \|z_k^{LU}\|$ and $\|\overline{D}_{k+1}^{\dagger} hr_k^{QR}\| = \|z_k^{QR}\|$. By the optimality conditions of Hybrid LSLU,

$$\|z_k^{LU}\| = \min_{x \in \mathcal{R}(L_k)} \left\| \begin{bmatrix} D_{k+1}^{\dagger} & 0 \\ 0 & L_k^{\dagger} \end{bmatrix} \left(\begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} A \\ \lambda I \end{bmatrix} \right) x \right\| = \min_{x \in \mathcal{R}(L_k)} \|\overline{D}_{k+1}^{\dagger} hr(x)\|. \quad (3.49)$$

Using (3.49) and the fact that x_k^{QR} is in $\mathcal{K}_k(A^T A, A^T b)$,

$$\|z_k^{LU}\| = \min_{x \in \mathcal{R}(L_k)} \|\overline{D}_{k+1}^{\dagger} hr(x)\| \leq \|\overline{D}_{k+1}^{\dagger} hr(x_k^{QR})\| = \|z_k^{QR}\|.$$

Thus

$$\|z_k^{LU}\| \leq \|z_k^{QR}\| = \|\overline{D}_{k+1}^{\dagger} hr_k^{QR}\| \leq \|\overline{D}_{k+1}^{\dagger}\| \|hr_k^{QR}\|.$$

Putting the above inequalities together gives the following relation,

$$\begin{aligned} \|hr_k^{LU}\| &= \|\overline{D}_{k+1} z_k^{LU}\| \\ &\leq \|\overline{D}_{k+1}\| \|z_k^{LU}\| \\ &\leq \|\overline{D}_{k+1}\| \|\overline{D}_{k+1}^{\dagger}\| \|hr_k^{QR}\|, \end{aligned}$$

so we conclude that $\|hr_k^{QR}\| \leq \|hr_k^{LU}\| \leq \kappa(\overline{D}_{k+1}) \|hr_k^{QR}\|$. □

To illustrate the behavior of the residual norms for Hybrid LSLU and Hybrid LSQR as well as to investigate the bound in Theorem 4, we plot in Figure 3.8 the residual norms per iteration for three different test problems: PRtomo, PRspherical, and PRseismic from the IR Tools package [16]. We fix $\lambda = 0.01$ and plot residual norms for Hybrid LSLU along with the lower and upper bounds from Theorem 4. We observe that the residual norms for Hybrid LSLU and Hybrid LSQR remain close for

PRtomo and PRspherical. As expected, the residual norms for solutions computed using Hybrid LSQR provide a lower bound for residual norms for solutions computed using Hybrid LSLU. The upper bound from Theorem 4 given by $\kappa(\overline{D}_{k+1})\|hr_k^{QR}\|$ becomes looser with more iterations. For details regarding the test problems, see Section 3.4.

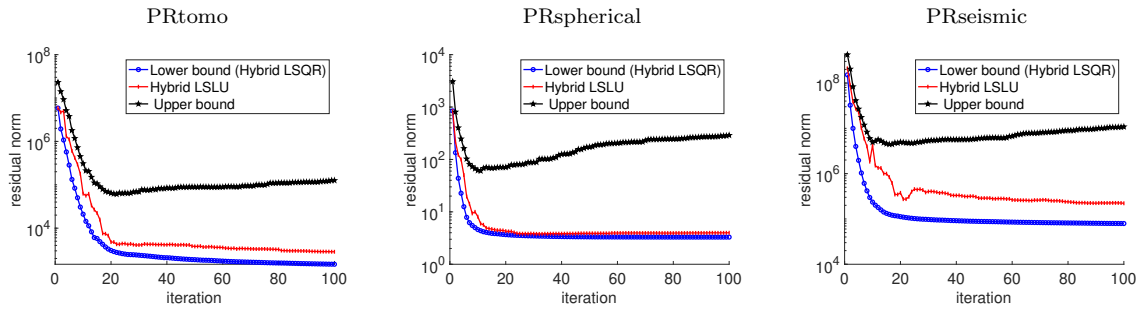

Figure 3.8: Residual norms per iteration for Hybrid LSLU, as well as corresponding bounds from Theorem 4. Note that the lower bound corresponds to Hybrid LSQR residual norms.

3.3.2 Computational Considerations

In this section we describe some of the computational aspects of Hybrid CMRH and Hybrid LSLU. In particular, we describe methods for selecting regularization parameter λ_k at each iteration in Section 3.3.2.1 and stopping criterion in Section 3.3.2.2.

3.3.2.1 Selecting Regularization Parameters

Our objective is to find an appropriate regularization parameter λ_k at each iteration of Hybrid CMRH and Hybrid LSLU that will not cause the regularized solution x_k to be overly oscillatory or too smooth. There are various approaches for selecting regularization parameters within hybrid projection methods [10]. We consider Tikhonov regularization for the projected problem (3.43). SVD based approaches can be used to find a good estimate for the regularization parameter λ_k, since the projected problem (3.43) is significantly smaller than (3.42).

Although not available in practice, we compute the optimal regularization parameter for simulated data to demonstrate the potential benefits of Hybrid CMRH and Hybrid LSLU. The optimal regularization parameter requires knowledge of the true solution and is obtained by minimizing the following expression:

$$\lambda_k = \arg\min_{\lambda} \|x_{\lambda,k} - x_{\text{true}}\|, \tag{3.50}$$

where $x_{\lambda,k}$ is the approximate solution at the kth iteration with the regularization parameter λ. Assume that $x_{\lambda,k} = x_0 + N_k y_{\lambda,k}$. Then (3.50) can be written as:

$$\min_{\lambda} \|x_{\lambda,k} - x_{\text{true}}\| = \min_{\lambda} \|x_0 + N_k y_{\lambda,k} - x_{\text{true}}\|. \tag{3.51}$$

From (3.51), we replace $y_{\lambda,k}$ with the solution to the normal equations of (3.43) to get

$$\min_{\lambda} \|x_0 + N_k (Z_{k+1,k}^T Z_{k+1,k} + \lambda^2 I)^{-1} Z_{k+1,k}^T \beta e_1 - x_{\text{true}}\| \tag{3.52}$$

and use the SVD of $Z_{k+1,k} = U_k \Sigma_k V_k^T$, to simplify (3.52)

$$\min_{\lambda} \|x_0 + N_k (V_k \Sigma_k^T \Sigma_k V_k^T + \lambda^2 I)^{-1} V_k \Sigma_k^T U_k^T \beta e_1 - x_{\text{true}}\|. \tag{3.53}$$

Thus, (3.53) is equivalent to

$$\min_{\lambda} \|x_{\lambda,k} - x_{\text{true}}\| = \min_{\lambda} \|x_0 + N_k V_k (\Sigma_k^T \Sigma_k + \lambda^2 I)^{-1} \Sigma_k^T U_k^T \beta e_1 - x_{\text{true}}\|. \tag{3.54}$$

We again emphasize that this is not a realistic regularization parameter choice criterion since it requires the knowledge of the true solution, but we can use it to demonstrate the performance of Hybrid CMRH and Hybrid LSLU.

In practice, techniques like the Generalized Cross Validation (GCV) method or the Discrepancy Principle can be implemented to estimate λ_k, at each iteration.

In this dissertation we focus on the GCV method, which is a predictive statistics-based approach where prior estimates of the error norm are not needed [23, 19]. Here, we assume that the regularization parameter λ_k should be able to predict any missing information. Although the GCV method is typically applied for the original problem, we follow a common approach in hybrid projection methods and use the GCV function for the projected problem (3.43), with matrix $Z_{k+1,k}$ from (3.43). The chosen regularization parameter minimizes the GCV function:

$$G_{Z_{k+1},\beta e_1}(\lambda) = \frac{k\|(I - Z_{k+1,k}Z_\lambda^\dagger)\beta e_1\|^2}{\text{trace}(I - Z_{k+1,k}Z_\lambda^\dagger)^2} \tag{3.55}$$

where $Z_\lambda^\dagger = (Z_{k+1,k}^T Z_{k+1,k} + \lambda^2 I)^{-1} Z_{k+1,k}^T$.

Using the SVD of $Z_{k+1,k}$, (3.55) can be rewritten as:

$$G_{Z_{k+1},\beta e_1}(\lambda) = \frac{k\beta^2 \left(\sum_{i=1}^{k} \left(\frac{\lambda^2}{\sigma_i^2 + \lambda^2} [U_k^T e_1]_i \right)^2 + \left(\left[U_k^T e_1 \right]_{k+1} \right)^2 \right)}{\left(1 + \sum_{i=1}^{k} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \right)^2}, \tag{3.56}$$

with the GCV parameter at the kth iteration being $\lambda_k = \arg\min_\lambda G_{Z_{k+1},\beta e_1}(\lambda)$.

The standard GCV function may not perform well for certain types of problems. For example, in statistical nonparametric modeling, the GCV function might choose parameters that are too small and thus produce a highly oscillatory approximate solution [11]. In our study, we find that the approximate solution is overly-smooth. To avoid this phenomenon, weighted-GCV (wGCV) is used, where the wGCV function

for the projected matrix $Z_{k+1,k}$ is defined as:

$$G(\omega,\lambda) = \frac{k\|(I - Z_{k+1,k}Z_\lambda^\dagger)\beta e_1\|^2}{(\text{trace}(I - \omega Z_{k+1,k}Z_\lambda^\dagger))^2} \tag{3.57}$$

$$= \frac{k\beta^2\left(\sum_{i=1}^{k}\left(\frac{\lambda^2}{\sigma_i^2+\lambda^2}\left[U_k^T e_1\right]_i\right)^2 + \left(\left[U_k^T e_1\right]_{k+1}\right)^2\right)}{\left(1+\sum_{i=1}^{k}\frac{(1-\omega)\sigma_i^2+\lambda^2}{\sigma_i^2+\lambda^2}\right)^2}. \tag{3.58}$$

Here, the denominator depends on a new parameter ω. Similar to the selection of the regularization parameter, we find that our choice of ω impacts the smoothness of the approximate solution. Thus, we must be careful in how we select the value for ω. If $\omega = 1$, then (3.57) becomes the standard GCV function (3.55). If $\omega > 1$, then we are subtracting a multiple of the filter factors thus producing less smooth solutions. Likewise if $\omega < 1$, then we are adding a multiple which produces smoother solutions. Therefore, we want to select the value of ω using the adaptive approach described in [11].

3.3.2.2 Stopping Criterion

Next we describe an approach to determine a suitable stopping criteria for Hybrid CMRH and Hybrid LSLU. Similar to the approach described in [11] and inspired by [4], we assume that λ is fixed and seek a stopping iteration k that minimizes a GCV function in terms of k,

$$\frac{n\|(I - AA_k^\dagger)b\|^2}{(\text{trace}(I - AA_k^\dagger))^2} \approx \frac{n\|M_{k+1}^\dagger(I - AA_k^\dagger)b\|^2}{(\text{trace}(I - AA_k^\dagger))^2} = \hat{G}(k), \tag{3.59}$$

where $A_k^\dagger$ is defined by considering the approximate solution produced by Hybrid CMRH or Hybrid LSLU, where, without loss of generality and to simplify the nota-

tion, we have considered $x_0 = 0$:

$$x_k = N_k y_{\lambda,k} = N_k Z_\lambda^\dagger M_{k+1}^\dagger b \equiv A_k^\dagger b.$$

If M_{k+1} contains orthonormal columns, the left-hand side of (3.59) can be simplified and computed exactly, as it is done in [11]. Here, we use the approximation:

$$\begin{aligned} n\|(I - AA_k^\dagger)b\| &\approx n\|M_{k+1}^\dagger(I - AA_k^\dagger)b\| \\ &= n\|(I - Z_{k+1,k}Z_\lambda^\dagger)M_{k+1}^\dagger b\| \end{aligned}$$

where $M_{k+1}^\dagger A N_k = M_{k+1}^\dagger M_{k+1} Z_{k+1,k} = Z_{k+1,k}$ and $M_{k+1}^\dagger b = \beta e_1$. Using the SVD of $Z_{k+1,k}$, the previous expression can be rewritten as:

$$n\|M_{k+1}^\dagger(I - AA_k^\dagger)b\|_2^2 = n\beta^2 \left(\left(\sum_{i=1}^{k} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + \left[U_k^T e_1\right]_i \right)^2 + \left(\left[U_k^T e_1\right]_{k+1} \right)^2 \right).$$

The denominator of (3.59) is equivalent to:

$$\begin{aligned} (\text{trace}(I - AA_k^\dagger))^2 &= (\text{trace}(I - AN_k Z_\lambda^\dagger M_{k+1}^\dagger))^2 \\ &= (\text{trace}(I - M_{k+1}Z_{k+1,k}Z_\lambda^\dagger M_{k+1}^\dagger))^2 \\ &= (\text{trace}(I) - \text{trace}(Z_{k+1,k}Z_\lambda^\dagger))^2 \\ &= \left((m-k) + \sum_{i=1}^{k} \frac{\lambda_k^2}{\sigma_i^2 + \lambda_k^2} \right)^2, \end{aligned}$$

where $m = n$ for Hybrid CMRH. Therefore, the left-hand side of (3.59) can be approximated by

$$\hat{G}(k) = \frac{n\beta^2 \left(\left(\sum_{i=1}^{k} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} + \left[U_k^T e_1 \right]_i \right)^2 + \left(\left[U_k^T e_1 \right]_{k+1} \right)^2 \right)}{\left((m-k) + \sum_{i=1}^{k} \frac{\lambda_k^2}{\sigma_i^2 + \lambda_k^2} \right)^2}. \tag{3.60}$$

$\hat{G}(k)$ is used to determine the stopping iteration, k. The algorithm will terminate when the difference between the values is small:

$$\left| \frac{\hat{G}(k+1) - \hat{G}(k)}{\hat{G}(1)} \right| < tol\,. \tag{3.61}$$

or when the minimum of $\hat{G}(k)$ continues to increase within a certain window of iterations.

3.4 Numerical Results

We now illustrate the effectiveness of CMRH, Hybrid CMRH, and Hybrid LSLU. First, we show a comparison between CMRH and other inner product free methods for square system matrices A with different properties. Next, we showcase its potential benefits with respect to other traditional methods in the simulated context of low precision arithmetic. We add a comparison between Hybrid CMRH and Hybrid GMRES for different image deblurring examples with multiple noise levels, and a seismic tomography simulated example to illustrate its applicability to large-scale problems. Finally, we conduct a comparison between Hybrid LSLU and Hybrid LSQR for different tomography examples. The system matrix, A, is rectangular. In all examples, white Gaussian noise was added to the measurements, and we define

the corresponding noise level as

$$\frac{\|e\|}{\|Ax_{\text{true}}\|} = nl.$$

This means that all elements of $e \in \mathbb{R}^m$ come from the same Gaussian distribution with zero mean and standard deviation [23, Section 3.5.1]. Computations for all numerical experiments and illustrations described in this study were done using MATLAB on an M-series MacBook Pro.

3.4.1 Comparison with other inner product free methods

In this section, we present a comparison between CMRH and other inner product free regularization methods: Landweber, Richardson (first-order), and Chebyshev (suitable for any non-singular symmetric or non-symmetric linear system). Note that in the case of very ill-posed problems, the latter is numerically equivalent to the second-order Richardson method [5, Section 7.2.5], which uses a fixed step length. This is due to the fact that the fraction between the summation and the subtraction of the upper and lower bounds of the eigenvalues is very close to 1 for ill-conditioned problems containing the system matrix of the normal equations.

In the following experiments, we present the relative error norm histories that correspond to three examples with different spectral properties. First, we reintroduce test problem Spectra, where $A \in \mathbb{R}^{64\times 64}$ is an SPD matrix and the smallest eigenvalues are numerically zero, making it very ill-conditioned. Next, we present an example where the system matrix is diagonally dominant, ill-conditioned, and tridiagonal. This is known as a Dorr matrix and it is from the MATLAB gallery. Here, $A \in \mathbb{R}^{256\times 256}$ is a non symmetric matrix with real positive eigenvalues. The chosen solution corresponds to the previously introduced Heat example from the Regularization Tools Package [25]. Finally, we examine a symmetric matrix with negative eigenvalues.

This matrix, where $A \in \mathbb{R}^{256\times256}$, violates the theoretical convergence conditions of Landweber, Richardson, and Chebyshev. The example corresponds to Deriv2 and it is from the Regularization Tools Package. In all scenarios, white Gaussian noise of appropriate level (e.g. `nl` $= 0.5\%$, `nl` $= 50\%$ and `nl` $= 0.1\%$) has been added to the measurements to achieve challenging but resolvable problems. For all comparison methods, bounds on either the eigenvalues of A (for symmetric matrices) or A^TA are required.

Assume s_1 and s_n are the smallest and largest eigenvalues of A. We select the bounds for the Chebyshev method to be $\min(s_1^2, s_n^2)$ and $1.001\min(s_1^2, s_n^2)$, which follows the implementation in [5, Section 7.2.5]. Similar to [5, Section 7.7.2], the step length for Landweber is $0.99/\max(s_1^2, s_n^2)$. Lastly, we select the step length for Richardson to be $0.99/(s_1+s_2)$, which follows the implementation in [5, Section 7.2.3]

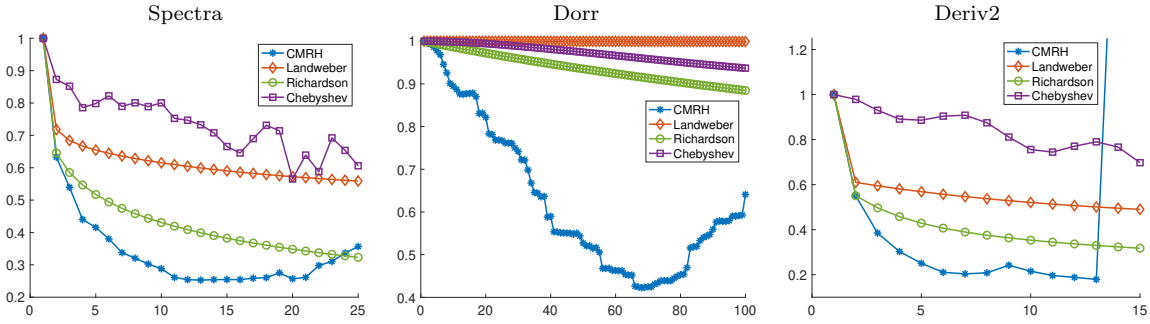

Figure 3.9: Relative error norm histories for problems with different spectral properties.

In Figure 3.9, it can be easily observed that CMRH is much faster than the other inner product free methods (particularly in the case of ill-posed problems). Moreover, CMRH does not require previous knowledge about the system matrix spectrum despite the price of increasing the memory requirements.

3.4.2 Results on low precision arithmetic

In this section we show a few instances of CMRH overcoming pathological behaviors arising when using GMRES in low precision arithmetic (low precision computations are simulated in MATLAB using Higham and Pranesh's `chop` function [28]). First, it should be noted that for large-scale inverse problems, such as those that arise in three-dimensional imaging applications, measured data contains noise, so one cannot expect to compute solutions to double-precision accuracy. Single-precision (32-bit) arithmetic generally provides sufficient accuracy and dynamic range; see, e.g., [1, 30, 31]. However, some challenges might arise when the precision is reduced further; in the following, we highlight such problems using two examples. For example, inner products (or norms) can fail in low precision due to under- or overflow when the length of the vectors increases. This can be observed in Figure 3.10, where relative error norm histories are presented when using two NVIDIA quarter precision formats [28]: test problem Deriv2 with $n = 4096$ using q52, which has 5 exponent and 2 significand bits (here we observe underflow while computing the norm of b), and test problem Shaw with $n = 6144$ using q43, which has 4 exponent and 3 significand bits (here we observe overflow while computing the norm of b). Test problems Deriv2 and Shaw are part of the Regularization Tools package [25].

Another major issue that can happen when using GMRES in low precision is that the basis vectors for the Krylov subspace can become orthogonal in the given precision, causing the algorithm to stop too early. This is illustrated in Figure 3.11, where we apply CMRH and GMRES to the Deriv2 test problem in half-precision arithmetic. In the left panel of Figure 3.11 one can observe the relative error norm histories for CMRH and GMRES, where GMRES stops at iteration 4. An easy explanation of this phenomenon can be observed in the right panel of Figure 3.11, where the first 20 diagonal elements of the Hessenberg matrix $H^A_{k+1,k}$ produced by GMRES (computed in double precision) are presented along with the machine precision (in

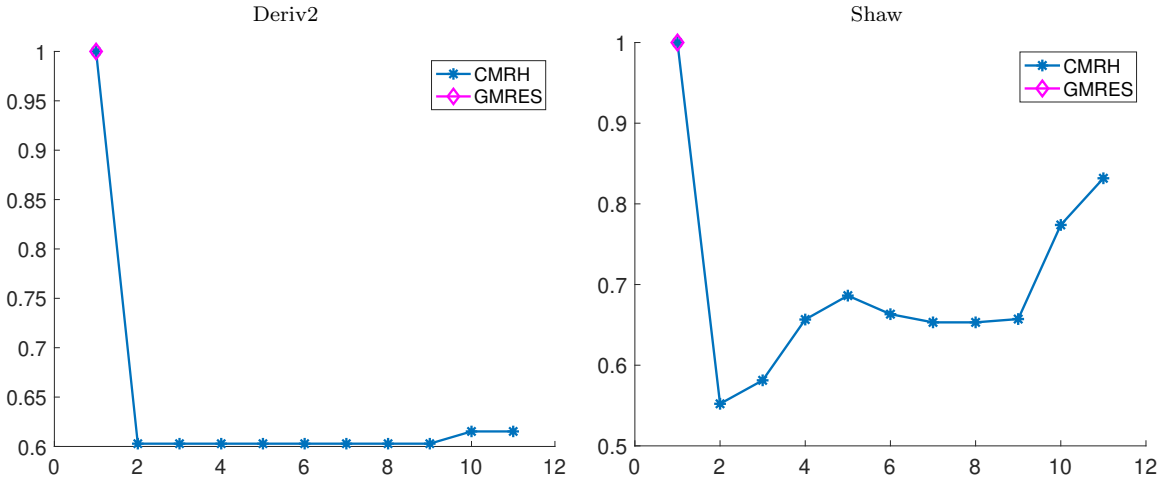

Figure 3.10: Relative error norm histories for Deriv2 and Shaw in precisions 'q52' and 'q43' (respectively). Early termination of GMRES is due to underflow (left) and overflow (right) computing vector norms.

half precision) in logarithmic scale. One can easily observe that, regardless of the particular implementation, the elements in the diagonal of $H^A_{k+1,k}$ quickly fall below machine precision, which prompts an early termination of the algorithm as previously observed.

3.4.3 Results on deblurring test problems.

To demonstrate the performance of Hybrid CMRH, we consider three test problems: PRblur, PRblurshake, and PRblurspeckle. In particular, PRblur represents Gaussian blur, PRblurshake simulates random motion blur (shaking), and PRblurspeckle models atmospheric blur. These problems are 2D deblurring problems from the IR Tools package [16] involving images with 256×256 pixels, corresponding to 65536×65536 system matrices A, and with added white Gaussian noise with noise levels between $nl = 10^{-3}$ and $nl = 10^{-1}$.

First, we show the blurry noisy images (with $nl = 10^{-2}$) along with the reconstructed images that were produced by Hybrid CMRH with the GCV scheme; see Figure 3.12. The reconstructions from Hybrid GMRES look very similar, as might

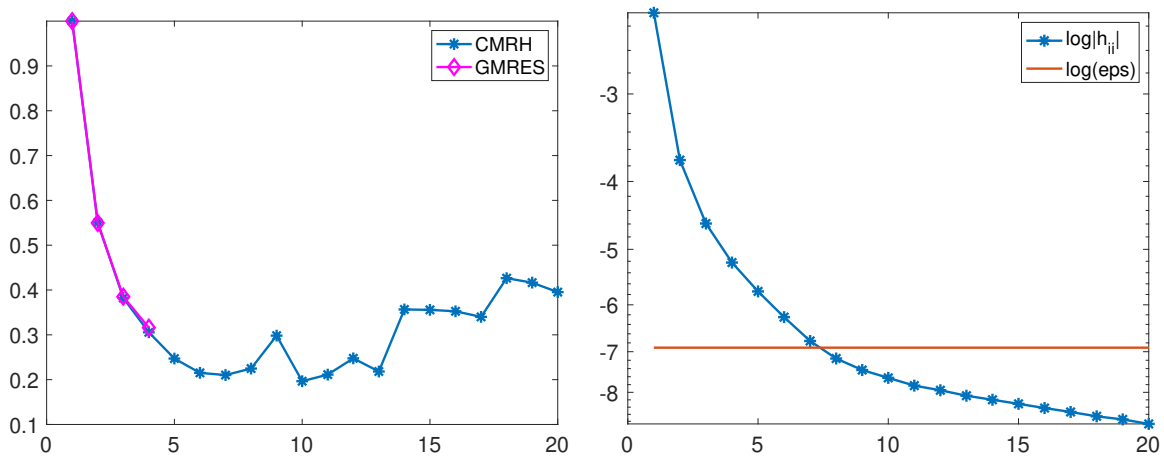

Figure 3.11: Test problem Deriv2. Left panel: relative error norm histories for GMRES and CMRH, where early stopping of GMRES is caused by numerical orthogonality of the basis vectors. Right panel: first 20 diagonal elements of the Hessenberg (Arnoldi) matrix produced by GMRES (computed in double precision) and machine (half) precision, in logarithmic scale.

be expected from the relative errors shown in Figure 3.13, so we do not include them here.

Next, we compare the relative error norms for Hybrid CMRH and Hybrid GMRES with GCV and optimal schemes for choosing the regularization parameter, which can be observed in Figure 3.13. Note that the star marker represents where the Hybrid CMRH algorithm stops based on the GCV stopping criteria in (3.60) and (3.61). In this figure, we observe that Hybrid CMRH and Hybrid GMRES behave in almost an identical manner for PRblurshake and PRblurspeckle. In PRblur, this is also true for the iterations before the stopping iteration.

Finally, additional experiments were performed with PRblur, PRblurshake, and PRblurspeckle using various noise levels. Table 3.1 displays the relative error norms for the solutions produced using the GCV stopping criteria for Hybrid CMRH and Hybrid GMRES. As proven in Section 3.3, the test results shown in Figures 3.12 and 3.13 and Table 3.1 illustrate that Hybrid CMRH and Hybrid GMRES have similar regularizing properties, while Hybrid CMRH is an inherent inner product free method.

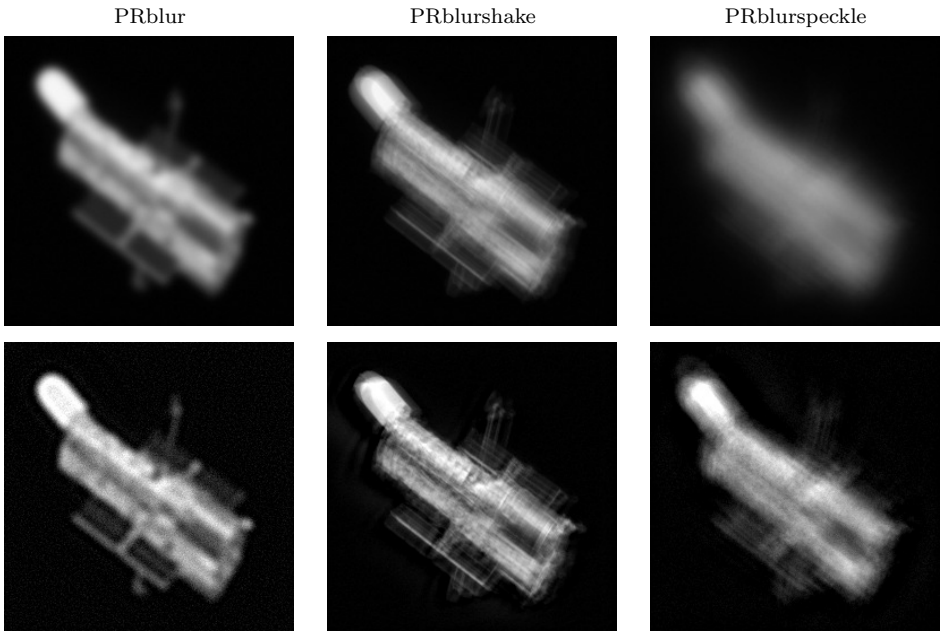

Figure 3.12: Top: Blurred and noisy measurements with $\texttt{nl} = 10^{-2}$. Bottom: reconstructed images using Hybrid CMRH.

3.4.4 Results on seismic test problem

In this section we present an additional large-scale problem from the AIR Tool II package [24], which can also be accessed via IR Tools as PRseismic. This is a simulated seismic travel-time tomography test problem. Since PRseismic produces a rectangular system, we apply Hybrid CMRH and Hybrid GMRES to the system matrix corresponding to the normal equations, so that $A \in \mathbb{R}^{65536\times 65536}$, where the images have 256×256 pixels.

Similarly to the image deblurring test problems in Section 3.4.3, we first show the measured noisy data, the true solution, and the reconstructed images that were produced by Hybrid CMRH and Hybrid GMRES with the GCV scheme at the stopping iteration; see Figure 3.14. Again, the reconstructions obtained using Hybrid GMRES and Hybrid CMRH have a similar quality. The relative error norm histories are shown

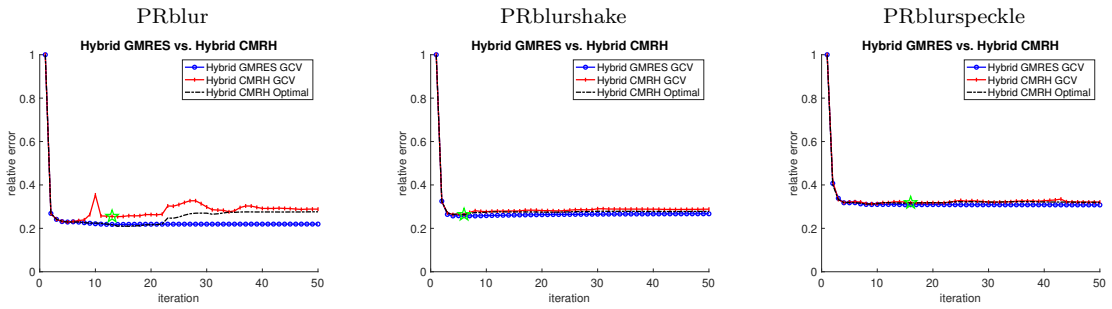

Figure 3.13: Relative error norm for 2D deblurring problems using Hybrid CMRH and Hybrid GMRES.

PRblur				
Method	Noise Level	Stopping Iteration	Reg Parameter	Relative Error
Hybrid CMRH	10^{-3}	8	5.5042×10^{-5}	0.2060
	10^{-2}	12	0.0589	0.2550
	10^{-1}	6	0.1396	0.3098
Hybrid GMRES	10^{-3}	14	0.0124	0.2016
	10^{-2}	12	0.0561	0.2179
	10^{-1}	5	0.1930	0.2493
PRblurshake				
Method	Noise Level	Stopping Iteration	Reg Parameter	Relative Error
Hybrid CMRH	10^{-3}	13	0.0051	0.2891
	10^{-2}	5	3.7149×10^{-5}	0.2631
	10^{-1}	5	3.7855×10^{-5}	0.3523
Hybrid GMRES	10^{-3}	4	0.1608	0.2565
	10^{-2}	4	0.1631	0.2581
	10^{-1}	3	0.2384	0.3310
PRblurspeckle				
Method	Noise Level	Stopping Iteration	Reg Parameter	Relative Error
Hybrid CMRH	10^{-3}	5	4.0987×10^{-5}	0.3167
	10^{-2}	15	0.0051	0.3177
	10^{-1}	5	3.5293×10^{-5}	0.7842
Hybrid GMRES	10^{-3}	5	0.0771	0.3153
	10^{-2}	5	0.0827	0.3181
	10^{-1}	9	0.2073	0.3414

Table 3.1: Numerical results for the three test problems PRblur, PRblurshake, and PRblurspeckle for various noise levels. Regularization parameters and relative error norms correspond to values at the stopping iteration.

in Figure 3.15. Although in this particular example, Hybrid CMRH terminates (at iteration 16) slightly before the optimal stopping iteration, this could be adjusted

by modifying tolerances used in the GCV criterion. The GCV stopping criteria for Hybrid GMRES (stopping at iteration 27) performed worse in this example.

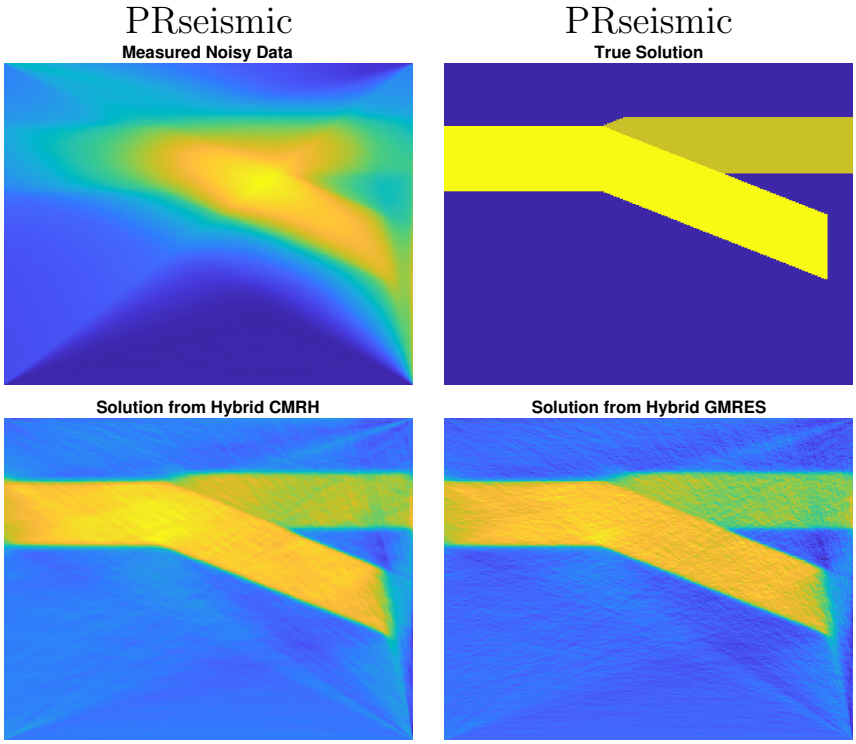

Figure 3.14: Measured noisy data (top left), true solution (top right), and reconstructed images using Hybrid CMRH and Hybrid GMRES (bottom row).

Additional experiments were conducted with different noise levels. Table 3.2 provides the relative error norm produced by Hybrid CMRH and Hybrid GMRES using the GCV stopping criteria. For `nl` $= 0.10\%$, Hybrid CMRH greatly outperforms Hybrid GMRES. Therefore, we find that Hybrid CMRH performs well when the noise level is increased.

3.4.5 Results on different tomography test problems

We now illustrate the effectiveness of Hybrid LSLU in comparison to Hybrid LSQR [41]. We use three different test problems: PRtomo, PRspherical, and PRseismic

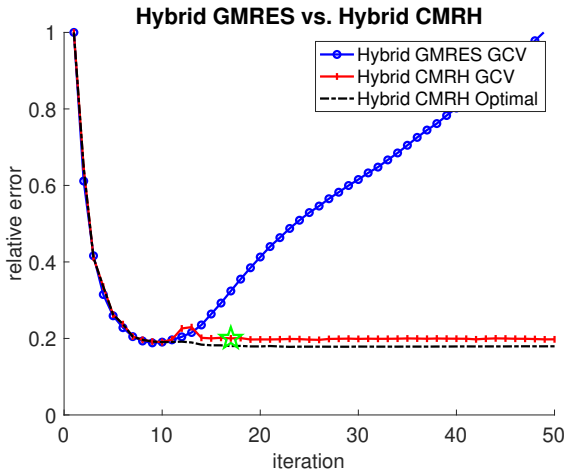

Figure 3.15: Relative error norm for seismic problem with `nl` $= 10^{-1}$ using Hybrid CMRH and Hybrid GMRES.

Numerical Results for PRseismic				
Method	Noise Level	Stopping Iteration	Reg Parameter	Relative Error
Hybrid CMRH	10^{-3}	39	0.0051	0.0947
	10^{-2}	9	6.1839×10^{-5}	0.1645
	10^{-1}	16	4.0069×10^{3}	0.2001
Hybrid GMRES	10^{-3}	17	220.394	0.1175
	10^{-2}	18	216.23	0.1196
	10^{-1}	27	121.54	0.5822

Table 3.2: Numerical results for PRseismic with various noise levels. Regularization parameters and relative errors correspond to values at the stopping iteration.

from the IR Tools package [16]. PRtomo generates data for x-ray tomographic reconstruction problems. PRspherical formulates a tomography test problem based on the spherical Radon transform where data consists of integrals along circles. This type of problem arises in photoacoustic imaging. PRseismic is the same test problem found in Section 3.4.4. These problems involve images with 256×256 pixels and correspond to matrix A that is 65160×65536 (PRtomo), 65522×65536 (PRspherical), and 131072×65536 (PRseismic) with $nl = 10^{-2}$. The noisy observations are provided in the top row of Figure 3.16.

We compute the reconstructed images for each problem using the proposed Hybrid LSLU method with wGCV to select the regularization parameter and GCV for the

stopping criterion. The reconstructions are provided in the middle row of Figure 3.16 with the true images provided in the bottom row for reference.

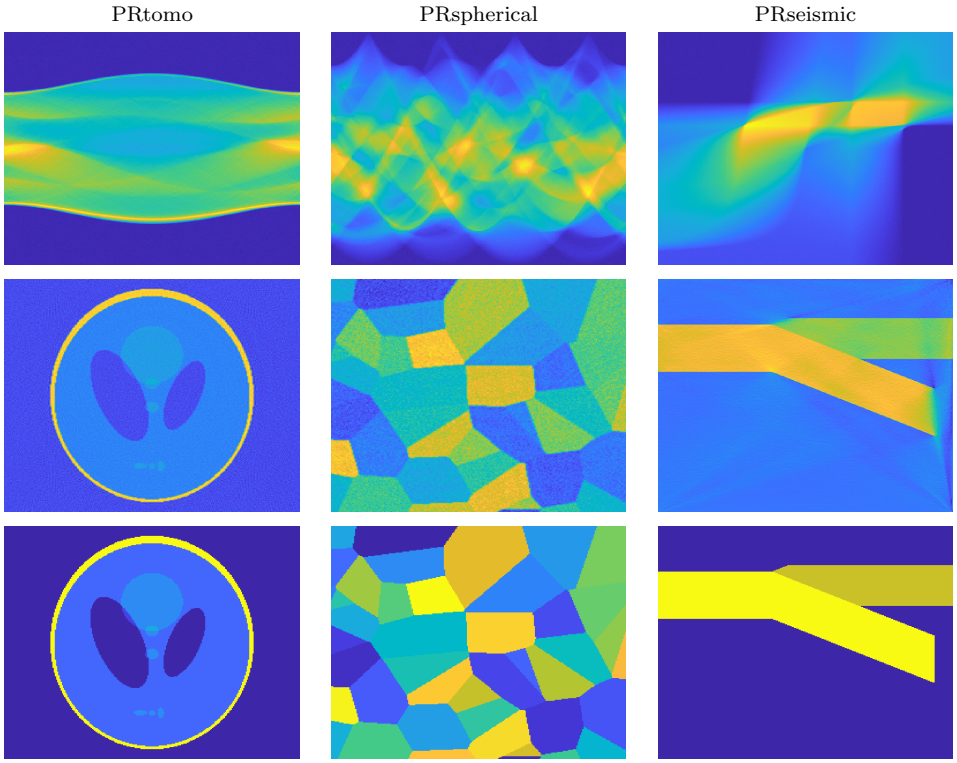

Figure 3.16: Measured noisy data, b (top row), reconstructed images using Hybrid LSLU (middle row), and true images, x_{true} (bottom row). The image proportions are accurate but, to aid visualization, the relative size between images is not.

Next, in Figure 3.17 we provide the relative reconstruction error norms per iteration of Hybrid LSLU with both the wGCV and optimal regularization parameter. Results for Hybrid LSQR with wGCV are provided for comparison. Hybrid LSQR selects ω by an approach described in [11]. From the Hybrid LSLU with optimal regularization plot, we observe that relative reconstruction error norms decrease and flatten at a nearly optimal value, which means that if a suitable choice of the regularization parameter is selected, Hybrid LSLU can provide a good regularized solution. The error is similar to that of Hybrid LSQR, and we remark that the stopping criteria for Hybrid LSLU performs well. These results demonstrate that Hybrid LSLU can provide comparable performance to Hybrid LSQR, with the same storage require-

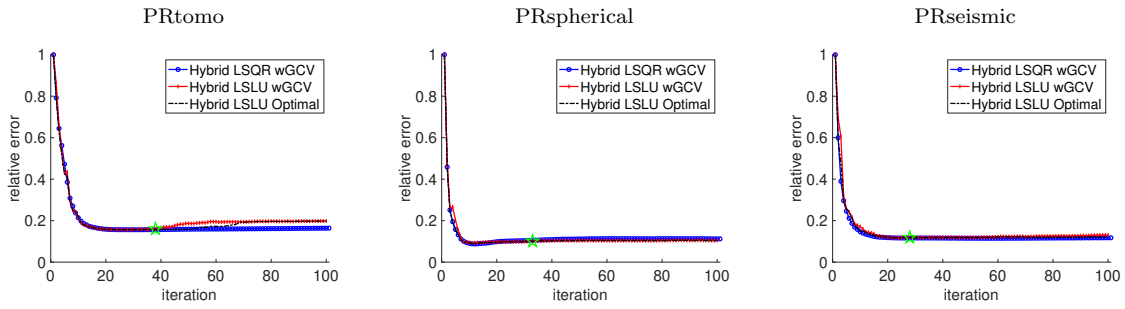

Figure 3.17: Relative reconstruction error norms per iteration of Hybrid LSLU with wGCV and the optimal regularization parameter. The automatically selected stopping iteration is highlighted with a star. Results for Hybrid LSQR with wGCV are provided for reference.

ments, lower computation cost and with the benefit of being inner product free.

The performance of Hybrid LSLU and Hybrid LSQR is similar for various noise levels. In Table 3.3, we provide the automatically selected stopping iteration, the computed regularization parameter using wGCV, the relative reconstruction error norm, for noise levels 10^{-3}, 10^{-2}, and 10^{-1}. We also give the iteration that produces the lowest value of the relative reconstruction error norm. We remark that the results for 10^{-2} are consistent with the results in Figure 3.17. We observe that for all noise levels, Hybrid LSLU and Hybrid LSQR perform comparably in all three test problems. This may be attributed to the stopping criteria and selected regularization parameter that result in similar reconstructions for both Hybrid LSLU and Hybrid LSQR.

Finally, for the PRseismic example, we investigate the images created by the basis vectors in Figures (3.18) and (3.19). Recall that in Hybrid LSLU, two sets of basis vectors are being constructed in an inner product free manner, one for each of the Krylov subspaces (3.21) and (3.22). We display 5 of the columns of the basis vectors from L_k and D_k reshaped into images of corresponding sizes for Hybrid LSLU, and provide the corresponding columns of the basis vectors generated via Hybrid LSQR for comparison. We observe that although the vectors generated via Hybrid LSLU

PRtomo						
Method	Noise	Stop Iter	Reg Param	Error	Best Iter	Best Error
Hybrid LSLU	10^{-3}	21	0.8043	0.1436	91	0.1195
	10^{-2}	38	1.6055	0.1598	28	0.1566
	10^{-1}	24	5.6387	0.5271	6	0.3536
Hybrid LSQR	10^{-3}	38	0.8300	0.1357	100	0.1184
	10^{-2}	31	3.0126	0.1561	29	0.1561
	10^{-1}	49	9.4351	0.4809	8	0.3426
PRspherical						
Method	Noise	Stop Iter	Reg Param	Error	Best Iter	Best Error
Hybrid LSLU	10^{-3}	18	0.0182	0.0624	55	0.0514
	10^{-2}	33	0.0420	0.1012	10	0.0928
	10^{-1}	31	0.3175	0.2803	6	0.1916
Hybrid LSQR	10^{-3}	53	0.0092	0.0515	100	0.0102
	10^{-2}	61	0.0347	0.1133	11	0.0877
	10^{-1}	11	0.1268	0.2983	4	0.1995
PRseismic						
Method	Noise	Stop Iter	Reg Param	Error	Best Iter	Best Error
Hybrid LSLU	10^{-3}	27	4.4428	0.1013	100	0.0816
	10^{-2}	28	3.4699	0.1188	21	0.1176
	10^{-1}	52	3.0218	0.5703	19	0.2185
Hybrid LSQR	10^{-3}	46	2.8851	0.0894	100	0.0839
	10^{-2}	22	12.8805	0.1166	56	0.1149
	10^{-1}	54	40.9810	0.2180	18	0.2122

Table 3.3: Numerical results for the three test problems PRtomo, PRspherical, and PRseismic for various noise levels. Regularization parameters and relative errors correspond to values at the stopping iteration. Noise is referring to noise level, and Error is referring to relative reconstruction error norm. Best Iter and Best Error correspond to the reconstruction with smallest relative reconstruction error norm.

at the kth iteration span the same subspace as the vectors generated via Hybrid LSQR, they have different features. We observe that the columns of L_k retrieve some characteristics of the true solution in early iterations; hence, we expect this to be a good basis for the solution. The columns of D_k contain information regarding the measurement $b \in \mathbb{R}^m$ or residual space, where the columns are basis vectors for $\mathcal{K}_k(AA^T, r_0)$. The ability of L_k to contain parts of the regularized solution is beneficial in helping to produce an accurate approximation of the true solution.

These images help to understand how different methods pick up different infor-

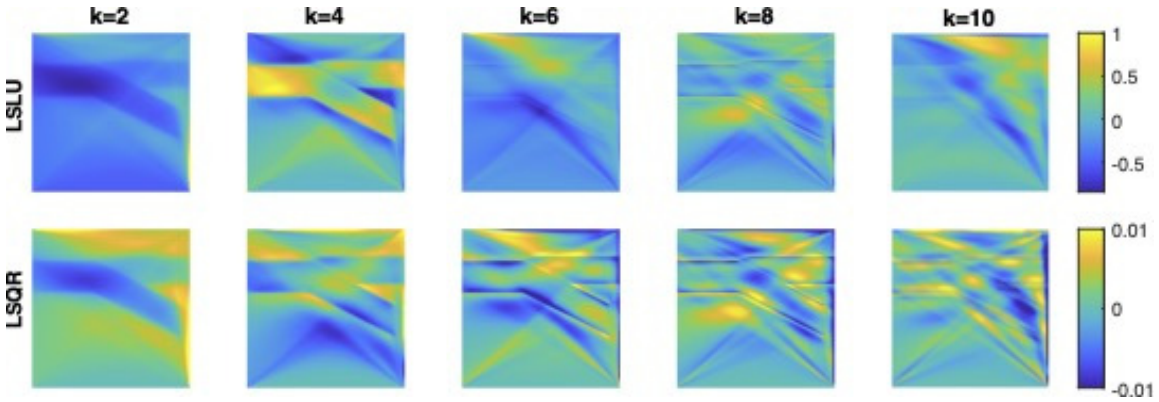

Figure 3.18: Basis vectors for the Krylov subspace (3.21) generated by LSLU and LSQR at iterations $k = 2, 4, 6, 8, 10$ for the PRseismic example.

mation. The LSQR basis vectors picks up the high-frequency information, due to the orthogonality requirement, and has a nice connection to SVD (frequency analysis). LSLU seems to pick up high-frequency information but does not project out previous vectors.

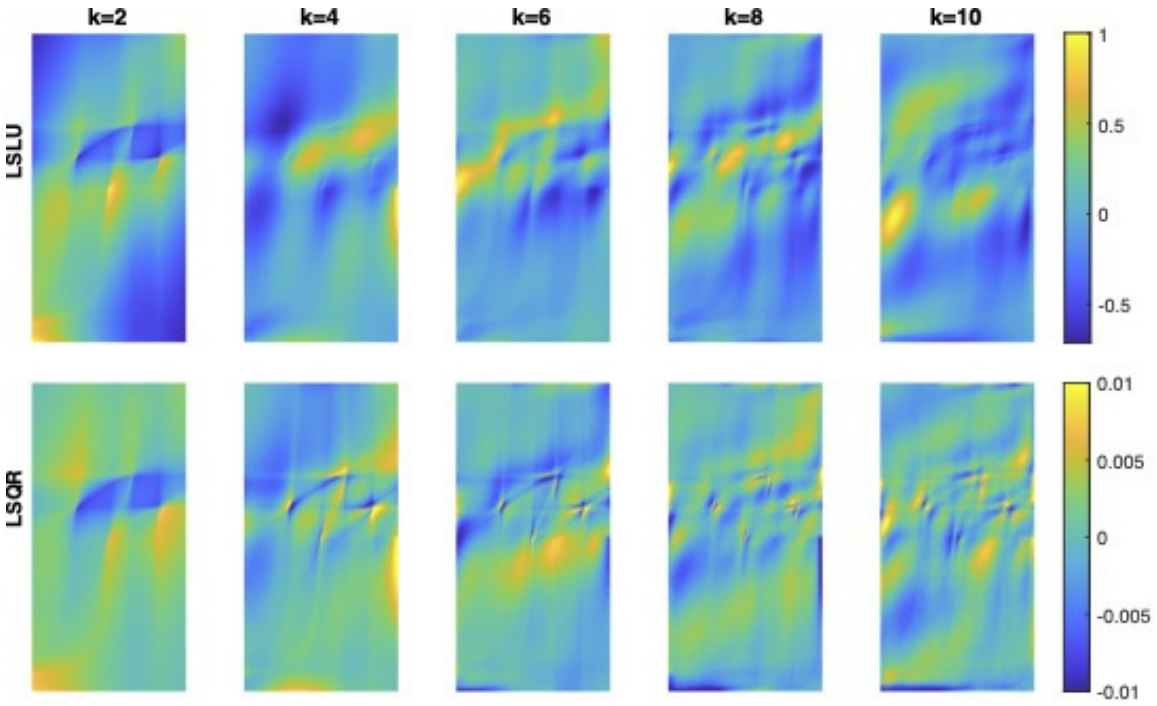

Figure 3.19: Basis vectors for the Krylov subspace (3.22) generated by LSLU and LSQR at iterations $k = 2, 4, 6, 8, 10$ for the PRseismic example.

Chapter 4

Sketched Inner Product Free Krylov Methods

This chapter, which is based on the collaborative work in [37], proposes a scheme that combines the inner product free Hessenberg projection for generating a solution subspace with a randomized sketch-and-solve approach.

In Chapter 3 we discussed CMRH and LSLU, which are quasi-minimum residual methods. These methods provided suitable solutions and various relationships between iterates [36]. However, in the context of inverse problems, the residual norm provides important information regarding the fit-to-data term, which depends on statistical assumptions about the measurement noise. In particular, the solution to the least squares problem (2.3) corresponds to a maximum likelihood estimate, and the solution to the Tikhonov problem (2.4) corresponds to a maximum posteriori estimate [9]. Thus, we seek inner product free Krylov methods that are also minimum residual methods. Unfortunately, this means that a tall, skinny least-squares problem needs to be solved at each iteration. We address this computational challenge by exploiting recent work on randomized methods, which produces solutions that can approximate closely the minimal residual norm at each iteration. The numerical results

will demonstrate that the proposed algorithm can solve large-scale inverse problems efficiently and without requiring inner products.

4.1 sCMRH and sLSLU

In this section, we propose new methods that use the (non-orthogonal) basis vectors for the relevant Krylov subspace, generated with the Hessenberg process (for square or rectangular systems), and approximately compute the solution of the projected least squares problem using a sketch-and-solve approach.

From Sections 3.1 and 3.2, we describe CMRH and LSLU as quasi-minimal residual methods that are free of inner products. At each iteration k, CMRH and LSLU are able to find an approximate solution of (2.3) by minimizing the following least squares problem:

$$\min_{x \in x_0 + \mathcal{R}(L_k)} \|M_{k+1}^{\dagger}(b - Ax)\|,$$

which is equivalent to solving

$$y_{\lambda,k} = \arg\min_{y \in \mathbb{R}^k} \|r_0 - AN_k y\|,$$

where $r_0 = b - Ax_0$ and $x_0 = 0$.

Now, assume S is an appropriate sketching matrix. We combine the projected least squares problem and the sketch-and-solve approach to form the following sketched projected problem:

$$\min_{y \in \mathbb{R}^k} \|S(b - AN_k y)\|. \tag{4.1}$$

The solution at the kth iteration is given by $x_k^{(S)} = N_k y_k^{(S)}$, where $y_k^{(S)}$ is the solution of (4.1). The crux of our approach lies in the assumption that at any given iteration, the solution $x_k^{(S)}$ will be close to the minimal residual norm $x_k^{(LS)}$ with high probability.

Using the subspace embedding property (2.7) on the residual norm, and recalling

that $x^{(LS)}$ is the minimal norm solution in the given Krylov subspace, i.e.,

$$\|b - Ax^{(LS)}\| = \min_{x \in x_0 + R(L_k)} \|b - Ax\|,$$

one can easily see that

$$\|b - Ax^{(LS)}\| \leq \|b - Ax_k^{(S)}\| \leq \frac{1+\epsilon}{1-\epsilon} \|b - Ax^{(LS)}\|. \tag{4.2}$$

For example, if $S \in \mathbb{R}^{s \times m}$ is a Gaussian sketch, (4.2) holds with a small probability of failure if $s \sim m \log(m)/\epsilon^2$ (in practice, this is usually further reduced to $s \sim m/\epsilon^2$). Moreover, as noted in [33], ϵ is the subspace embedding constant of the matrix $[A\ b]$ (i.e. appending b to the matrix A). This satisfies $\frac{1+\epsilon}{1-\epsilon} = \kappa_2(SQ^{[A\,b]})$, where $Q^{[A\,b]}$ is the orthogonal matrix obtained from the QR factorization of $[A\ b]$, and ϵ is usually of the order of 0.5. An accurate mathematical description of the relevant theory can be found in [32, Section 8.7.], and sharper bounds with a different probability of failure can be found in the seminal paper [40].

This means that, in theory one can pick ϵ to be small enough so that the solution $x_k^{(S)}$ produced by either sketched CMRH (sCMRH) or sketched LSLU (sLSLU) will have a smaller residual norm than that of the solution x_k obtained using the inner product free (but not sketched) counterparts. Although this is a very strong motivation of this work, it cannot, however, always be guaranteed in practice.

Moreover, we know that if S is a Gaussian sketch, then the sketch and solve solution to the projected problem is an unbiased estimate for the least-squares solution. Consider the projected problem and the sketched, projected problem:

$$\min_{y \in \mathbb{R}^k} \|b - AN_k y\| \quad \text{and} \quad \min_{y \in \mathbb{R}^k} \|S(b - AN_k y)\|,$$

respectively, where AN_k is a full column rank matrix. Then

$$\mathbb{E}[(SAN_k)^\dagger(Sb)] = (AN_k)^\dagger b$$

where † denotes the Moore-Penrose pseudoinverse. Since N_k is independent of the sketch, we have that $\mathbb{E}[x_k^{(S)}] = \mathbb{E}[N_k(SAN_k)^\dagger(Sb)] = N_k\mathbb{E}[(SAN_k)^\dagger(Sb)] = N_k(AN_k)^\dagger b = x_k^{(LS)}$. In addition, we can further analyze the expected squared residual norm as

$$\begin{aligned}\mathbb{E}[\|b - AN_k y_k^{(S)}\|] &= \left(1 + \frac{k}{s-k-1}\right) \min_y \|b - AN_k y\| \\ &= (1+\varepsilon) \min_y \|b - AN_k y\|\end{aligned}$$

when the sketch dimension is $s = \frac{k}{\varepsilon} + k + 1$. See [13, 14] and references therein.

Algorithms 9 and 10 provide the details for sCMRH and sLSLU. Note that for sCMRH and sLSLU, one sketching matrix is needed $S \in \mathbb{R}^{s\times m}$; however, to incorporate Tikhonov regularization, we will need two sketching matrices, $S_1 \in \mathbb{R}^{s\times n}$ and $S_2 \in \mathbb{R}^{s\times m}$.

As can be observed in Algorithm 10, the generalized Hessenberg method with partial pivoting requires locating the largest absolute value of r_0, v_0, and two other vectors at each iteration, which can be costly due to global communications. To avoid this occurrence, we apply the pivoting alternative described in Section 3.2.1 where only a small subset of elements in those vectors is observed, and the pivot is taken to be the element with the largest absolute value from this subset. In practice, this leads to a reduced communication cost and a stable way of building the non-orthogonal basis for the relevant Krylov Subspace methods: in our experiments, we never observed the condition number of the basis to grow significantly large. However, this strategy might still increase the condition number $\kappa(\overline{D}_{k+1})$ for LSLU or $\kappa(\overline{L}_{k+1})$ for CMRH, rendering the solutions from LSLU and CMRH less accurate. Therefore,

Algorithm 9: sCMRH

Require: A, b, x_0, maxiter, S

Define $p = [1, 2, \ldots, n]^T$,
$r_0 = b - Ax_0$
Determine i such that $|r_0(i)| = \|r_0\|_\infty$
$\beta = r_0(i)$; $l_1 = r_0/\beta$; $p(1) \Leftrightarrow p(i)$
for $k = 1, \ldots,$ maxiter **do**
  $u = Al_k$
  $z_k = S * u$
  for $j = 1, \ldots, k$ **do**
    $H(j,k) = u(p(j))$; $u = u - H(j,k)l_j$
  end for
  if $k < m$ and $u \neq 0$ **then**
    Determine $i \in \{k+1, \ldots, m\}$ such that $|u(p(i))| = \|u(p(k+1:m))\|_\infty$
    $H(k+1,k) = u(p(i))$; $d_k = u/H(k+1,k)$; $p(k+1) \Leftrightarrow p(i)$
  else
    $H(k+1,k) = 0$;
  end if
  Compute y_k to be the minimizer of $\|Sr_0 - SAL_k y\|_2^2 = \|Sr_0 - Zy\|_2^2$
  $x_k = x_0 + L_k y_k$
end for

this is a particular case of LSLU and CMRH where the sketched methods have a great potential compared to their non-sketched counterparts. All numerical results will implement this technique for different sample sizes.

From (3.38), we derived a bound on the difference between the residual norms of solutions computed using LSLU and LSQR. It was shown that if the condition number of $\hat{R}_{k+1}$, the upper triangular matrix from the QR decomposition of D_{k+1}, does not grow too quickly, then the residual norms associated with the approximate solutions of LSLU and LSQR at each iteration are close to each other. In Section 4.3, we will compare the residual norm of sLSLU with the residual norms from (3.38).

In the next section we propose a sketch-and-solve approach to project the Tikhonov problem on a Krylov subspace and approximately compute the solution of the least squares problem (4.1).

Algorithm 10: sLSLU

Require: A, b, x_0, maxiter, S_1, S_2

Define $t = [1, 2, \ldots, m]^T$, $g = [1, \ldots, n]^T$.
$r_0 = b - Ax_0$
Determine i such that $|r_0(i)| = \|r_0\|_\infty$
$\beta = r_0(i)$; $d_1 = r_0/\beta$; $t(1) \Leftrightarrow t(i)$
$v_0 = A^T r_0$
Determine i_2 such that $|v_0(i_2)| = \|v_0\|_\infty$
$\alpha = v_0(i_2)$; $l_1 = v_0/\alpha$; $g(1) \Leftrightarrow g(i_2)$
$r = A^T d_1$; $W(1,1) = r(g(1))$
for $k = 1, \ldots,$ maxiter **do**
  $u = A^T d_k$
  $z_k = S_2 * u$
  for $j = 1, \ldots, k$ **do**
    $H(j,k) = u(t(j))$; $u = u - H(j,k)d_j$
  end for
  if $k < m$ and $u \neq 0$ **then**
    Determine $i \in \{k+1, \ldots, m\}$ such that $|u(t(i))| = \|u(t(k+1:m))\|_\infty$
    $H(k+1,k) = u(t(i))$; $d_k = u/H(k+1,k)$; $t(k+1) \Leftrightarrow t(i)$
  else
    $H(k+1,k) = 0$;
  end if
  $q = A^T d_k$
  for $j = 1, \ldots, k$ **do**
    $W(j,k+1) = q(g(j))$; $q = q - W(j,k+1)l_j$
  end for
  if $k < n$ and $q \neq 0$ **then**
    Determine $i_2 \in \{k+1, \ldots, n\}$ such that $|q(g(i_2))| = \|q(g(k+1:n))\|_\infty$
    $W(k+1,k+1) = u(g(i_2))$; $l_{k+1} = q/W(k+1,k+1)$; $g(k+1) \Leftrightarrow g(i_2)$
  else
    break
  end if
  Compute y_k to be the minimizer of $\|S_2 r_0 - S_2 A L_k y\|_2^2 = \|S_2 r_0 - Zy\|_2^2$
  $x_k = x_0 + L_k y_k$
end for

4.2 Extensions for Tikhonov Regularization

Recall the standard Tikhonov regularization problem (2.4). As stated in Section 3.2, the Hybrid LSLU method computes a solution at the kth iteration to the following

optimization problem

$$\min_{x \in x_0 + \mathcal{R}(L_k)} \|D_{k+1}^{\dagger}(b - Ax)\|^2 + \lambda^2 \|L_k^{\dagger} x\|^2, \tag{4.3}$$

which is equivalent to solving

$$y_{\lambda,k} = \arg\min_{y \in \mathbb{R}^k} \|\beta e_1 - H_{k+1,k} y\|^2 + \lambda^2 \|y\|^2, \tag{4.4}$$

where β is selected using the sampling strategy. Similarly, one can use a sketch-and-solve approach to project the Tikhonov problem on a Krylov subspace and approximately compute the solution of the least squares problem using the following expression:

$$\min_{y \in \mathbb{R}^k} \|S_2(b - AL_k y)\|^2 + \lambda^2 \|S_1(L_k y)\|^2, \tag{4.5}$$

where S_1 and S_2 are appropriate sketching matrices. Similar to LSQR and LSLU, we derived a bound on the difference between the residual norms of solutions computed using LSLU and LSQR for the Tikhonov problem. It was shown that if the condition number of $\bar{D}_{k+1}$ does not grow too quickly, then the residual norms associated to the solution of LSLU for the Tikhonov problem is close to the residual norm of the solution obtained with LSQR for the Tikhonov problem (see Theorem 4). To display behavior of sLSLU with Tikhonov regularization, we will plot the associated residual norm against the residuals norms denoted in Theorem 4 (see Section 4.3). An implementation of sLSLU with Tikhonov regularization is provided in Algorithm 12, which corresponds to Algorithm 10 if $\lambda = 0$. Note that for all numerical results, we select a fixed regularization parameter. The corresponding implementation of sCMRH with Tikhonov regularization can be seen in Algorithm 11.

Algorithm 11: sCMRH with Tikhonov Regularization

Require: A, b, x_0, maxiter, S_1, S_2, λ

Define $p = [1, 2, \ldots, n]^T$,
$r_0 = b - Ax_0$
Determine i such that $|r_0(i)| = \|r_0\|_\infty$
$\beta = r_0(i)$; $l_1 = r_0/\beta$; $p(1) \Leftrightarrow p(i)$
for $k = 1, \ldots,$ maxiter **do**
  $u = Al_k$
  $z_k = S_2 * u$
  for $j = 1, \ldots, k$ **do**
    $H(j,k) = u(p(j))$; $u = u - H(j,k)l_j$
  end for
  if $k < m$ and $u \neq 0$ **then**
    Determine $i \in \{k+1, \ldots, m\}$ such that $|u(p(i))| = \|u(p(k+1:m))\|_\infty$
    $H(k+1,k) = u(p(i))$; $d_k = u/H(k+1,k)$; $p(k+1) \Leftrightarrow p(i)$
  else
    $H(k+1,k) = 0$;
  end if
  Compute $y_{\lambda_k,k}$ to be the minimizer of $\|S_2 r_0 - S_2 A L_k y\|^2 + \lambda^2 \|S_1 L_k y\|^2$
  $x_k = x_0 + L_k y_{\lambda_k,k}$
end for

Algorithm 12: sLSLU with Tikhonov Regularization

Require: A, b, x_0, maxiter, S_1, S_2, λ

Define $t = [1, 2, \ldots, m]^T$, $g = [1, \ldots, n]^T$.
$r_0 = b - Ax_0$
Determine i such that $|r_0(i)| = \|r_0\|_\infty$
$\beta = r_0(i)$; $d_1 = r_0/\beta$; $t(1) \Leftrightarrow t(i)$
$v_0 = A^T r_0$
Determine i_2 such that $|v_0(i_2)| = \|v_0\|_\infty$
$\alpha = v_0(i_2)$; $l_1 = v_0/\alpha$; $g(1) \Leftrightarrow g(i_2)$
$r = A^T d_1$; $f_1 = S_1 * r$; $W(1,1) = r(g(1))$
for $k = 1, \ldots,$ maxiter **do**
  $u = A^T d_k$
  $z_k = S_2 * u$
  for $j = 1, \ldots, k$ **do**
    $H(j,k) = u(t(j))$; $u = u - H(j,k)d_j$
  end for
  if $k < m$ and $u \neq 0$ **then**
    Determine $i \in \{k+1, \ldots, m\}$ such that $|u(t(i))| = \|u(t(k+1:m))\|_\infty$
    $H(k+1,k) = u(t(i))$; $d_k = u/H(k+1,k)$; $t(k+1) \Leftrightarrow t(i)$
  else
    $H(k+1,k) = 0$;
  end if
  $q = A^T d_k$
  $f_{k+1} = S_1 q$
  for $j = 1, \ldots, k$ **do**
    $W(j,k+1) = q(g(j))$; $q = q - W(j,k+1)l_j$
  end for
  if $k < n$ and $q \neq 0$ **then**
    Determine $i_2 \in \{k+1, \ldots, n\}$ such that $|q(g(i_2))| = \|q(g(k+1:n))\|_\infty$
    $W(k+1,k+1) = u(g(i_2))$; $l_{k+1} = q/W(k+1,k+1)$; $g(k+1) \Leftrightarrow g(i_2)$
  else
    break
  end if
  Compute $y_{\lambda_k,k}$ to be the minimizer of $\|S_2 r_0 - S_2 A L_k y\|^2 + \lambda^2 \|S_1 L_k y\|^2$
  $x_k = x_0 + L_k y_{\lambda_k,k}$
end for

4.3 Numerical Results

In this section, we illustrate the effectiveness of sketched inner product free Krylov methods: sketched CMRH (sCMRH) and sketched LSLU (sLSLU) in comparison to

their non-sketched counterparts CMRH and LSLU, and the classical GMRES and LSQR. We utilize three different test problems: a deblurring problem, a neutron tomography simulation from the IR Tools package [16], and an example with real data consisting of two open access datasets from the Finnish Inverse Problems Society [8, 21]. Moreover, for Tikhonov regularization with fixed λ, we provide numerical results for sLSLU, LSLU, and LSQR for some of these test problems. Note that each sketch matrix has the following dimensions: $S_1 \in \mathbb{R}^{s\times n}$, $S_2 \in \mathbb{R}^{s\times m}$ where $s = 10 * (max\ iteration + 1)$.

4.3.1 Deblurring problem

The first experiment consists of a deblurring problem, where the aim is to reconstruct MATLAB's test image 'cameraman' of size 256×256 pixels, which was corrupted by motion blur and additive Gaussian noise. The forward model was simulated using IR Tools [16], and noise was added so that the observation contained a 1% noise level.

Since this is a square problem, we compare sCMRH to CMRH and GMRES. The relative reconstruction error norm and residual norms per iteration are displayed in Figure 4.1. We observe that the curves corresponding to sCMRH closely resembles that of GMRES, as dictated by the theory, while the residual norms for CMRH deviate as the iterations proceed. This can also be observed in the relative error norms. Thus, sCMRH produces solutions that more closely resemble GMRES solutions, but without the need for inner product computations. The reconstructed images are shown in Figure 4.2.

4.3.2 Neutron tomography simulation

Neutron imaging is a tomography technique based on gamma-rays that allows us to inspect the interior of dense or metallic objects. This is because, contrary to x-ray tomography, the absorption of neutrons is higher in 'light' elements and lower in metallic

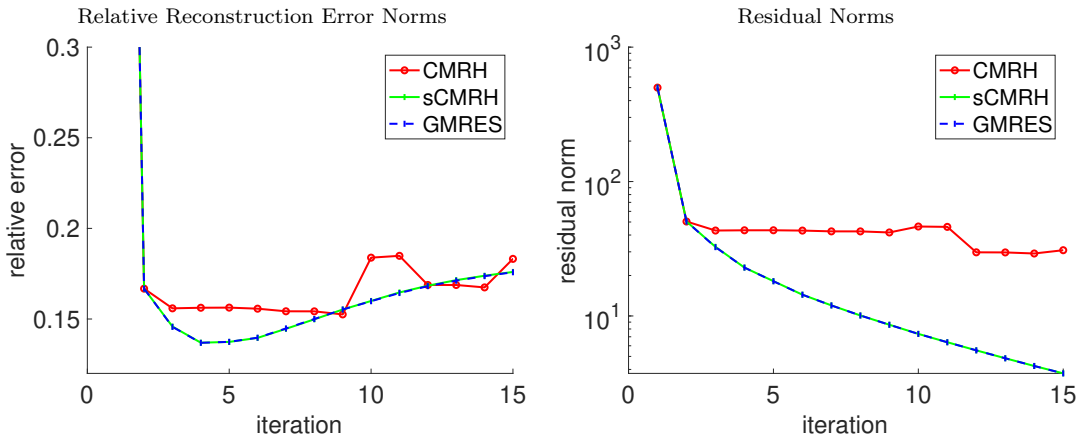

Figure 4.1: Relative reconstruction error norms (left) and residual norms (right). In this case, sketched CMRH uses the pivots dictated by the maximum absolute value from a set of randomly sampled coefficients (5).

elements. See, for example, the interior of a padlock in [3]. Note that, mathematically, this CT modality has the same mathematical model as x-ray tomography, but using different absorption coefficients for each material. Since most datasets for this type of tomography are proprietary, in this example, we use MATLAB's built-in demo image 'circuit.tiff', which has similar structure to neutron tomography examples.

The sketched LSLU (sLSLU) algorithm implements the pivoting alternative described in Section 3.2.1. The sample size to approximate the infinity norm contains 25 entries. Note that varying the sample size does not appear to drastically change the numerical results. In Figure 4.3, we provide the relative reconstruction error norms per iteration of sLSLU. Results for LSQR and LSLU are provided for comparison. We

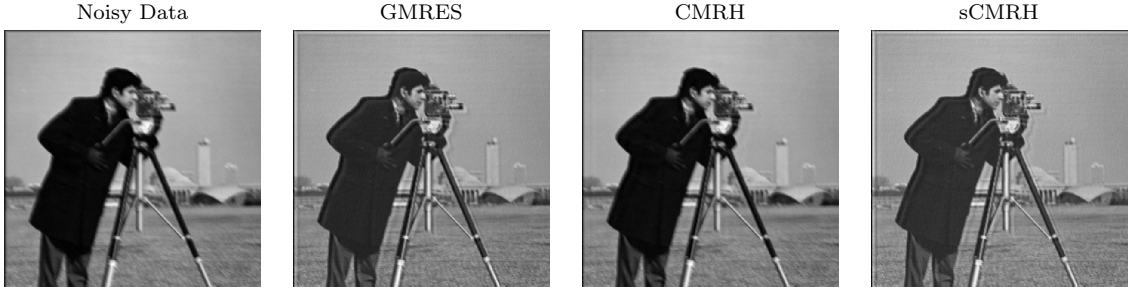

Figure 4.2: Measured noisy data, and reconstructed images using GMRES, CMRH, and sCMRH.

observe that sLSLU performs better than LSLU and similar to LSQR, especially in the earlier iterations. Provided we implement a "good" stopping criteria, we can compute an approximation that is of comparable quality to that produced with LSQR, while avoiding inner products.

The residual norms are also plotted for comparison (see right plot in Figure 4.3). In Figure 4.3, we find that residual norms for sLSLU closely follow the lower bound, which corresponds to residual norms for LSQR. Similar to the relative error plot, we find that the behavior of sLSLU aligns with LSQR. The reconstructions where the relative error is the smallest are provided in Figure 4.4.

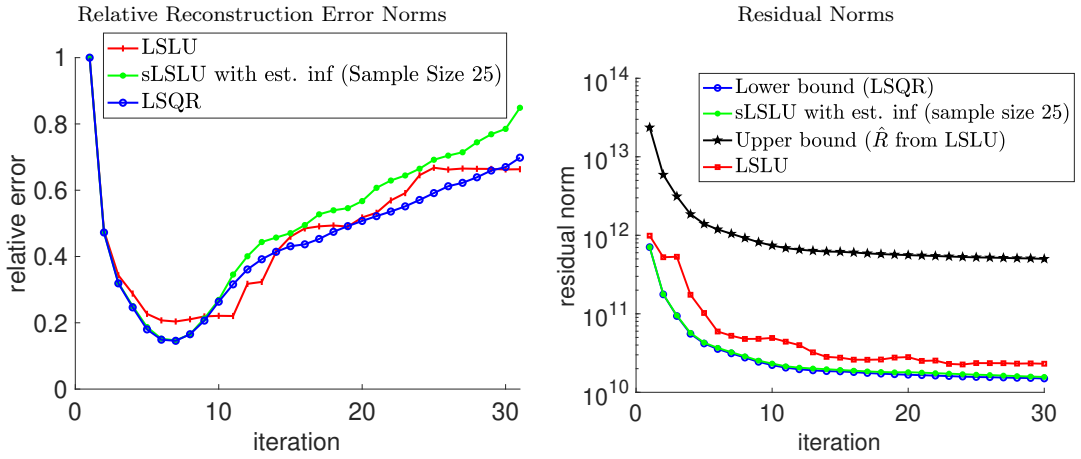

Figure 4.3: Relative reconstruction error norms (left) and residual norms (right). In this case, sketched LSLU uses the pivots dictated by the maximum absolute value from a set of randomly sampled coefficients (25).

Next we consider the performance of sLSLU for the Tikhonov problem. We fix $\lambda = 26$, and we plot the residual norm of sLSLU compared to LSLU and LSQR in Figure 4.5. We also provide the relative reconstruction error norms per iteration. We observe that the inclusion of the regularization term stabilizes the semi-convergence for all methods. For the Tikhonov problem, sLSLU mirrors the behavior of LSQR. An adaptive approach to find a "good" regularization parameter during the iterations is a topic of future work. Image reconstructions corresponding to 30 iterations are provided in Figure 4.6.

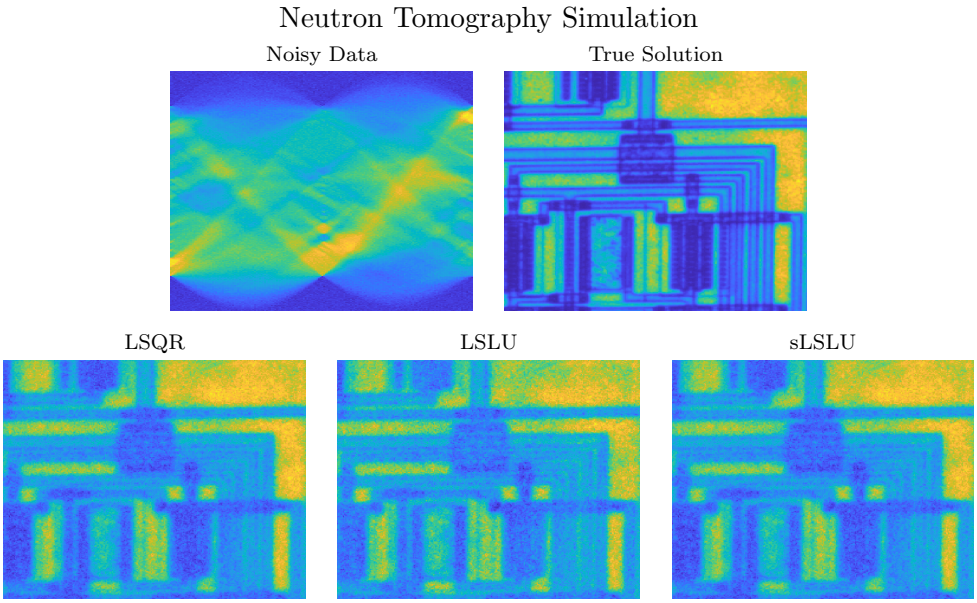

Figure 4.4: Measured noisy data, true solution, and reconstructed images from LSQR, LSLU, and sLSLU. The image proportions are accurate but, to aid visualization, the relative size between images is not.

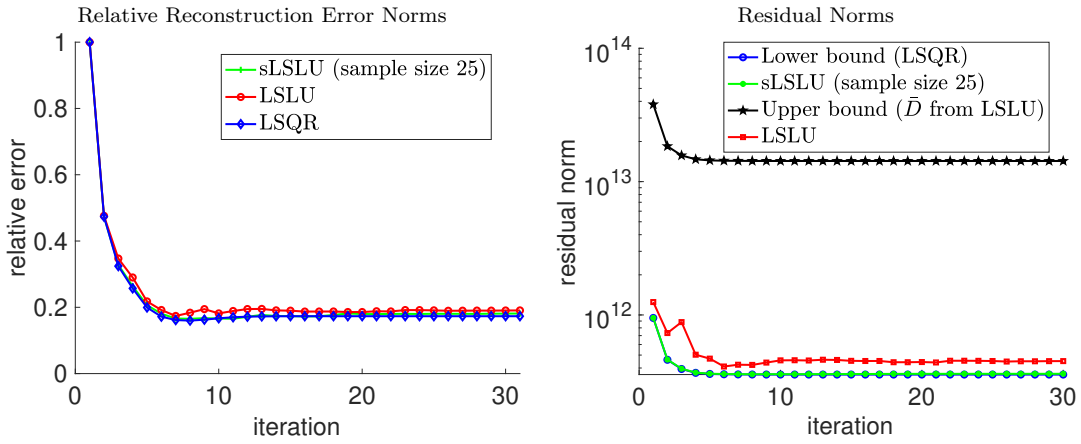

Figure 4.5: Relative reconstruction error norms (left) and residual norms (right). In this case, sketched LSLU with Tik. Reg. uses pivots dictated by the maximum absolute value from a set of randomly sampled coefficients (25).

4.3.3 Real data examples

The Finnish Inverse Problems Society has provided the following open access datasets: a tomographic x-ray of carved cheese and a walnut. Both datasets consist of x-ray sinograms where each sinogram is obtained by fan-beam projection. The observed

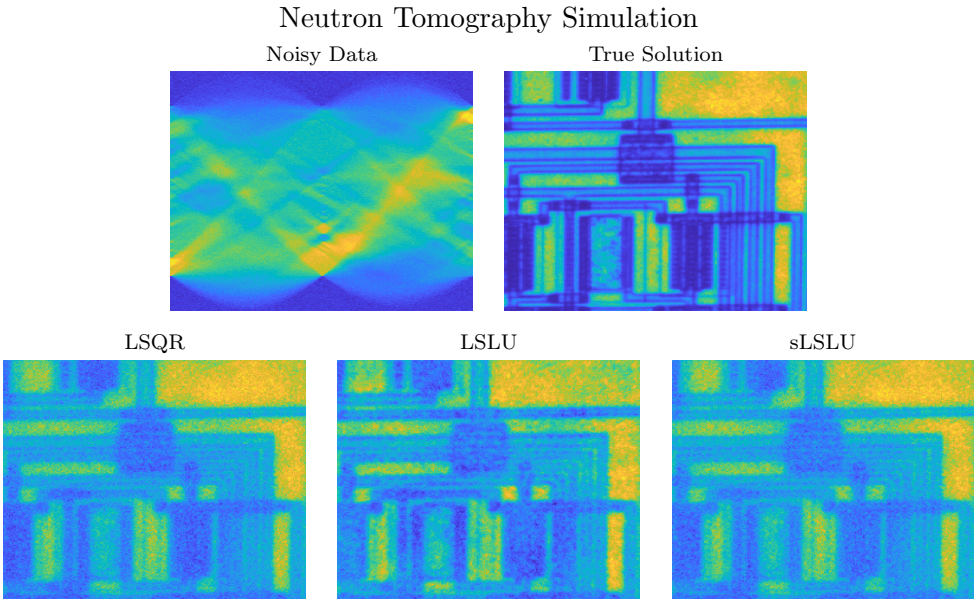

Figure 4.6: Measured noisy data, true solution, and reconstructed LSQR, LSLU, and sLSLU for solving the Tikhonov problem. The image proportions are accurate but, to aid visualization, the relative size between images is not.

data for the carved cheese dataset containing 360 projections and the walnut dataset containing 120 projections are provided in Figures 4.8 and 4.9 respectively. For these problems, there is no true image, so we rely on residual norms per iteration to compare algorithms.

To illustrate the behavior of the residual norms for sLSLU, LSLU, and LSQR as well as the bounds in Section 3.2.2, we plot in Figure 4.7 the residual norms per iteration for the carved cheese and walnut datasets. The samples size to approximate the infinity norm contains 25 entries. For both datasets, we observe that the residual norms for sLSLU and LSQR remain close together, with the residual norms for LSLU being a bit larger. Therefore, we may expect that the approximate solutions from sLSLU should mirror those from LSQR. This is verified in Figure 4.8 and Figure 4.9, where the reconstructed images using LSQR, LSLU, and sLSLU are provided. All reconstructions correspond to iteration 30.

Finally, we consider sLSLU with Tikhonov regularization for these examples, where we plot the the residual norm of sLSLU with Tikhonov regularization in Fig-

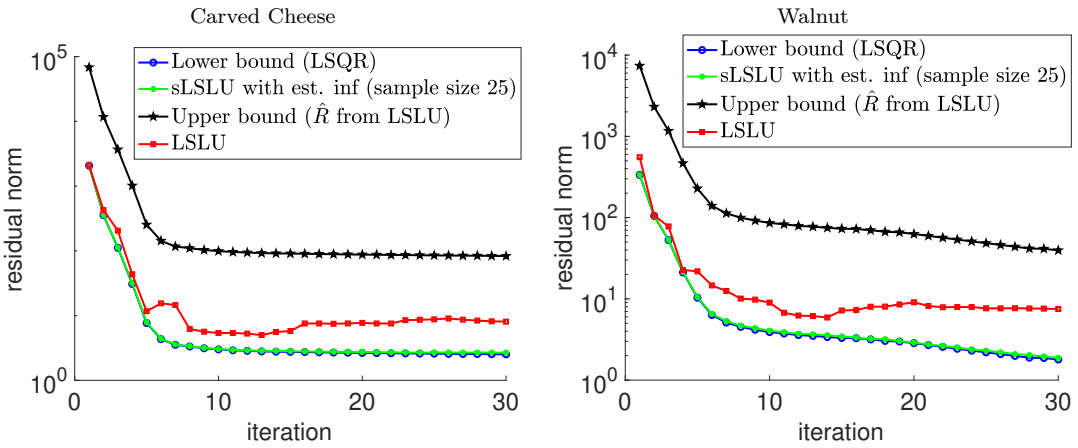

Figure 4.7: Residual norms per iteration for sLSLU and LSLU, as well as corresponding bounds from Theorem 2.1 of [6]. Note that the lower bound corresponds to LSQR residual norms.

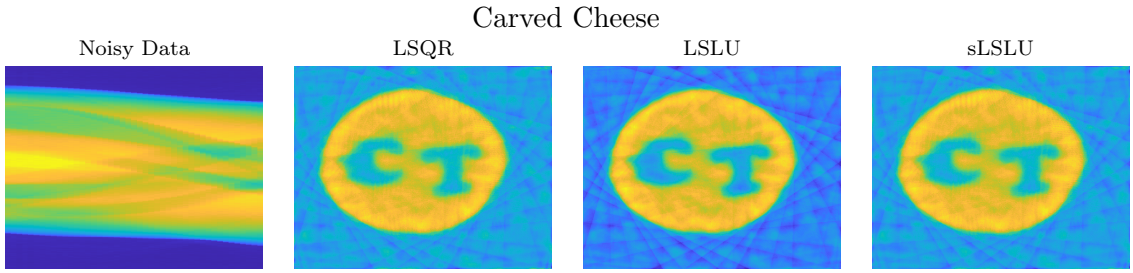

Figure 4.8: Measured noisy data, and reconstructed images from LSQR, LSLU, and sLSLU. The image proportions are accurate but, to aid visualization, the relative size between images is not.

ure 4.10. We fix $\lambda = 1$ for both problems. Similar to the nonregularized problems, the residual norms for sLSLU closely follow the lower bound, which corresponds to residual norms for LSQR. Thus, we may expect that provided we have a "good" estimate for the regularization parameter, sLSLU with Tikhonov regularization will produce a better approximation of the solution than LSLU on the Tikhonov problem. We also provide reconstructed images for both datasets in Figure 4.11 and Figure 4.12.

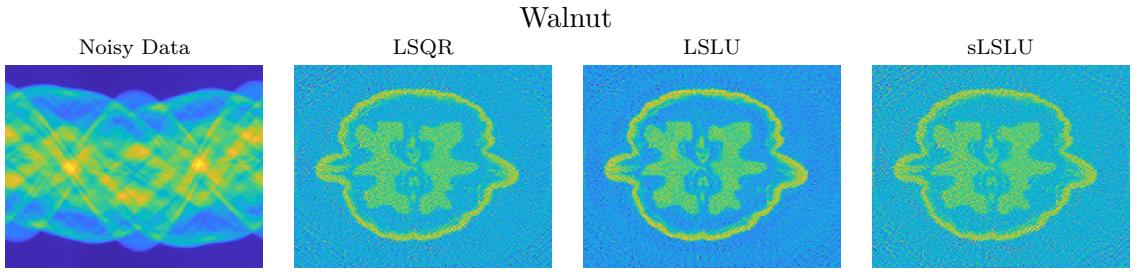

Figure 4.9: Measured noisy data, and reconstructed images from LSQR, LSLU, and sLSLU. The image proportions are accurate but, to aid visualization, the relative size between images is not.

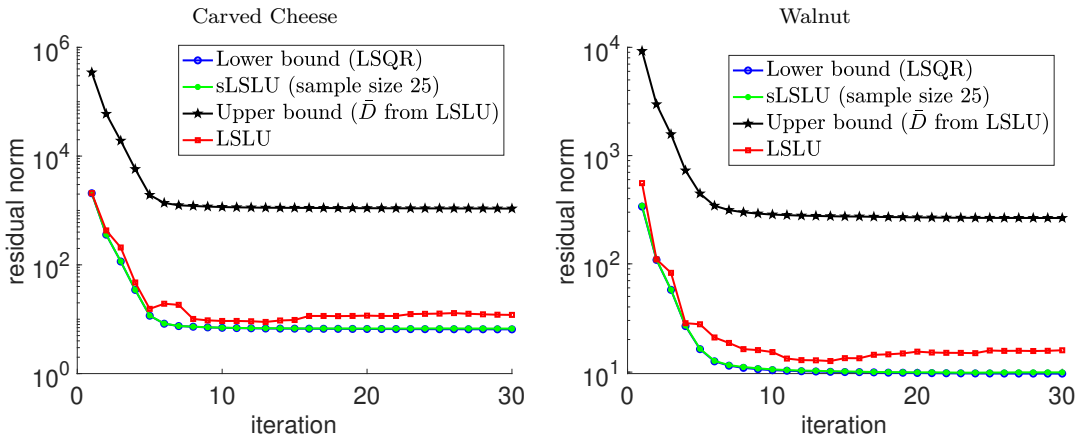

Figure 4.10: Residual norms per iteration for the Tikhonov problem correspond to sLSLU, LSLU, as well as corresponding bounds from Theorem 3.1 of [6]. Note that the lower bound corresponds to LSQR. The regularization parameter $\lambda = 1$.

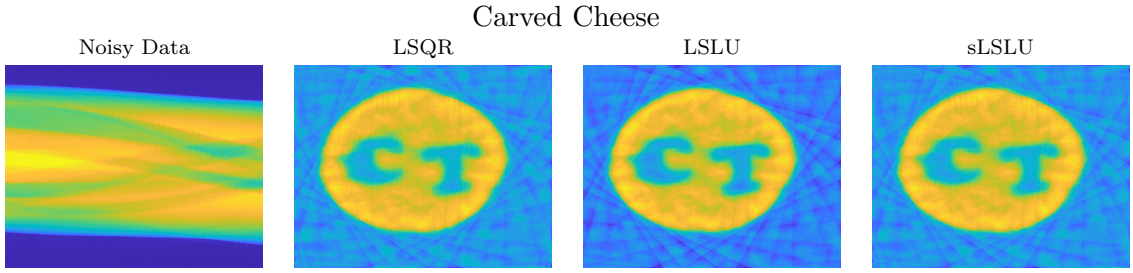

Figure 4.11: Measured noisy data, and Tikhonov reconstructions for LSQR, LSLU, and sLSLU. The image proportions are accurate but, to aid visualization, the relative size between images is not.

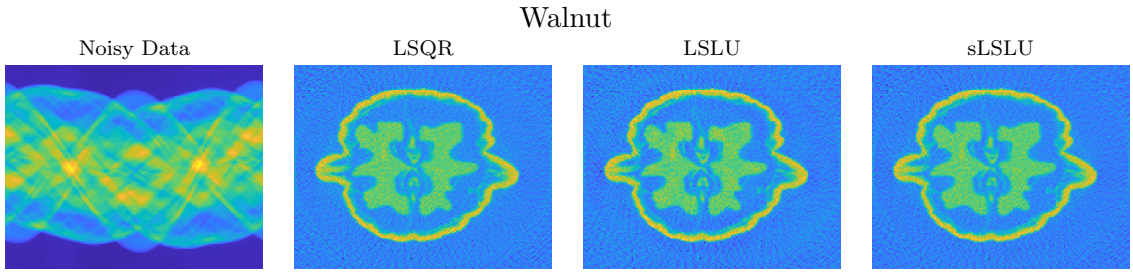

Figure 4.12: Measured noisy data, and Tikhonov reconstructions for LSQR, LSLU, and sLSLU. The image proportions are accurate but, to aid visualization, the relative size between images is not.

Chapter 5

Conclusion

In this dissertation we presented several new inner product free Krylov subspace methods for large-scale linear inverse problems. We began with establishing CMRH as a robust iterative regularization method, expanding its potential to a diverse range of applications in large-scale ill-posed problems. Through a detailed study, we offer both theoretical insights and a deeper understanding of the properties of the projected problems involved. The regularization characteristics of CMRH are shown to effectively filter solutions, a conclusion that we support using empirical evidence. Once this method is proven to be apt for ill-posed problems, we show that CMRH is able to deliver a solution of similar quality much faster than other inner product free alternatives, and highlight the advantages of using inner product free methods in low precision arithmetic scenarios, where CMRH overcomes certain limitations of GMRES. Moreover, we introduce a novel hybrid version of the CMRH method (Hybrid CMRH), the first hybrid method to be inner product free. We again stress that it was first necessary to demonstrate the regularization capabilities of CMRH before proposing the hybrid version, Hybrid CMRH. The performance of CMRH and Hybrid CMRH is validated through its application to various ill-posed problems.

Next, we introduced two new inner product free Krylov methods for rectangular

large-scale linear ill-posed inverse problems. Based on our numerical observations, the Hybrid LSLU method is comparable to Hybrid LSQR in its ability to select regularization parameters during the iterative process and stabilize semiconvergence. Both LSLU and Hybrid LSLU only require matrix-vector multiplications with A and its transpose, making them appealing for large-scale problems.

Lastly, we introduced two new inner product free Krylov methods, sCMRH and sLSLU, that incorporate randomization techniques for solving large-scale linear inverse problems. Both methods are based on the Hessenberg method with partial pivoting for building bases that span Krylov subspaces, and hence do not require inner product computations (e.g., orthogonalizations). They also exploit randomized sketching to solve the projected problems, thereby producing solutions with a smaller residual norm compared to existing inner product free Krylov methods. Numerical experiments show that the performance of sCMRH is comparable to that of GMRES and that the performance of sLSLU is comparable to that of LSQR. Moreover, sCMRH and sLSLU have smaller residual norm solutions, compared to CMRH and LSLU respectively. The sketched Krylov methods can be adapted to incorporate Tikhonov regularization provided that an appropriate regularization parameter is selected. Since sCMRH and sLSLU are all inner product free, they may be useful in solving problems with mixed-precision and parallel computing, which is a topic of future work.

Bibliography

[1] C. K. Anand. Robust solvers for inverse imaging problems using dense single-precision hardware. *J. Math. Imaging Vis.*, 33:105–120, 2009.

[2] Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1994. doi: 10.1137/1.9781611971538.

[3] Ander Biguri, Tomoyuki Sadakane, Reuben Lindroos, Yi Liu, Malena Sabaté Landman, Yi Du, Manasavee Lohvithee, Stefanie Kaser, Sepideh Hatamikia, Robert Bryll, Emilien Valat, Sarinrat Wonglee, Thomas Blumensath, and Carola-Bibiane Schönlieb. Tigre v3: Efficient and easy to use iterative computed tomographic reconstruction toolbox for real datasets. *ArXiv preprint*, 2024. URL `https://arxiv.org/abs/2412.10129`.

[4] Åke Björck, Eric Grimme, and Paul Van Dooren. An implicit shift bidiagonalization algorithm for ill-posed systems. *BIT Numerical Mathematics*, 34(4): 510–534, 1994.

[5] Åke Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, 1996. doi: 10.1137/1.9781611971484.

[6] Ariana N. Brown, Julianne Chung, James G. Nagy, and Malena Sabaté Landman. Inner product free Krylov methods for large-scale inverse problems, 2024. URL `https://arxiv.org/abs/2409.05239`.

[7] Ariana N. Brown, Malena Sabaté Landman, and James G. Nagy. H-CMRH: An inner product free hybrid Krylov method for large-scale inverse problems. *SIAM Journal on Matrix Analysis and Applications*, 46(1):232–255, 2025. doi: 10.1137/24M1634874.

[8] Tatiana A Bubba, Markus Juvonen, Jonatan Lehtonen, Maximilian März, Alexander Meaney, Zenith Purisha, and Samuli Siltanen. Tomographic x-ray data of carved cheese. *arXiv preprint arXiv:1705.05732*, 2017.

[9] Daniela Calvetti and Erkki Somersalo. *An introduction to Bayesian scientific computing: ten lectures on subjective computing*, volume 2. Springer Science & Business Media, New York, 2007.

[10] Julianne Chung and Silvia Gazzola. Computational methods for large-scale inverse problems: a survey on hybrid projection methods. *SIAM Review*, 66(2): 205–284, 2024.

[11] Julianne Chung, James G Nagy, and Dianne P O'Leary. A weighted GCV method for Lanczos hybrid regularization. *Electronic Transactions on Numerical Analysis*, 28:149–167, 2008.

[12] Julianne Chung, Sarah Knepper, and James G. Nagy. Large-scale inverse problems in imaging. In Handbook of Mathematical Methods in Imaging, O. Scherzer (Ed.). Springer, New York, NY, 2015.

[13] Michał Dereziński and Michael W Mahoney. Recent and upcoming developments in randomized numerical linear algebra for machine learning. In *Proceedings of*

the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 6470–6479, 2024.

[14] Petros Drineas, Michael W Mahoney, Shan Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, 2011.

[15] D. K. Faddeev and V. N. Faddeeva. *Computational Methods of Linear Algebra*. Freeman, San Francisco and London, 1963. Translated by R. C. Williams.

[16] S Gazzola, P C Hansen, and J G Nagy. IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems. *Numerical Algorithms*, pages 1–39, 2018.

[17] Silvia Gazzola, Paolo Novati, and Maria Rosaria Russo. On Krylov projection methods and Tikhonov regularization. *Electronic Transactions on Numerical Analysis*, 44:83–123, 2015.

[18] G. H. Golub and R. S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. Parts I and II. *Numerische Mathematik*, 3:147–156, 1961.

[19] Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[20] Stefan Güttel and Igor Simunec. A sketch-and-select arnoldi process. *SIAM Journal on Scientific Computing*, 46(4):A2123–C533, 2024. doi: 10.1137/23M1588007.

[21] Keijo Hämäläinen, Lauri Harhanen, Aki Kallonen, Antti Kujanpää, Esa Niemi,

and Samuli Siltanen. Tomographic x-ray data of a walnut. *arXiv preprint arXiv:1502.04064*, 2015.

[22] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, Philadelphia, 1997.

[23] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. SIAM, Philadelphia, 2010.

[24] P. C. Hansen and Jørgensen. AIR Tools II: algebraic iterative reconstruction methods, improved implementation. *Numerical Algorithms*, 79:107–137, 2018.

[25] Per Christian Hansen. Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems. *Numerical algorithms*, 6(1):1–35, 1994.

[26] Per Christian Hansen, James G. Nagy, and Dianne P. O'Leary. *Deblurring Images*. Society for Industrial and Applied Mathematics, 2006. doi: 10.1137/1.9780898718874.

[27] Mohammed Heyouni and Hassane Sadok. A new implementation of the CMRH method for solving dense linear systems. *Journal of Computational and Applied Mathematics*, 213:387–399, 2008.

[28] N. J. Higham and S. Pranesh. Simulating low precision floating-point arithmetic. *SIAM Journal on Scientific Computing*, 41(5):C585–C602, 2019.

[29] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, 1989.

[30] H. Li. Double precision is not necessary for LSQR for solving discrete linear ill-posed problems. *J. Sci. Comput.*, 98:55, 2024.

[31] C. Maaß and SSaer, M. and Kachelrieß. CT image reconstruction with half precision floating-point values. *Medical Physics*, 38:S95–S105, 2011.

[32] Per-Gunnar Martinsson and Joel A. Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020. doi: 10.1017/S0962492920000021.

[33] Maike Meier, Yuji Nakatsukasa, Alex Townsend, and Marcus Webb. Are sketch-and-precondition least squares solvers numerically stable? *SIAM Journal on Matrix Analysis and Applications*, 45(2):905–929, 2024. doi: 10.1137/23M1551973.

[34] Christopher C Paige and Michael A Saunders. Algorithm 583: LSQR: Sparse linear equations and least squares problems. *ACM Transactions on Mathematical Software (TOMS)*, 8(2):195–209, 1982.

[35] Youcef Saad. Krylov subspace methods on supercomputers. *SIAM Journal on Scientific and Statistical Computing*, 10(6):1200–1232, 1989. doi: 10.1137/0910073. URL `https://doi.org/10.1137/0910073`.

[36] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003. ISBN 0-89871-534-2. doi: 10.1137/1.9780898718003.

[37] Malena Sabaté Landman, Ariana N. Brown, Julianne Chung, and James G. Nagy. Randomized and inner-product free Krylov methods for large-scale inverse problems, 2025. URL `https://arxiv.org/abs/2502.02721`.

[38] Hassane Sadok. CMRH: A new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm. *Numerical Algorithms*, 20:303–321, 1999.

[39] Hassane Sadok and Daniel B. Szyld. A new look at CMRH and its relation to GRMES. *BIT Numerical Mathematics*, 52:485–501, 2012.

[40] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 143–152, 2006. doi: 10.1109/FOCS.2006.37.

[41] Michael A Saunders and Christopher C Paige. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8, 1982.

[42] H. Joel Trussell. Convergence criteria for iterative restoration methods. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31(1):129–136, 02 1983.

[43] Curt R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2002.

[44] Michael Zhdanov. *Geophysical Inverse Theory and Regularization Problem*, volume 36. Elsevier, 01 2002. ISBN 0444510893.