FAST AND ROBUST SOLUTION TECHNIQUES FOR LARGE SCALE LINEAR LEAST SQUARES PROBLEMS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

İbrahim Kurban Özaslan

July 2020

FAST AND ROBUST SOLUTION TECHNIQUES FOR LARGE SCALE LINEAR LEAST SQUARES PROBLEMS
By İbrahim Kurban Özaslan
July 2020

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Orhan Arıkan(Advisor)

Sinan Gezici

Elif Vural

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

FAST AND ROBUST SOLUTION TECHNIQUES FOR LARGE SCALE LINEAR LEAST SQUARES PROBLEMS

İbrahim Kurban Özaslan
M.S. in Electrical and Electronics Engineering
Advisor: Orhan Arıkan
July 2020

Momentum Iterative Hessian Sketch (M-IHS) techniques, a group of solvers for large scale linear Least Squares (LS) problems, are proposed and analyzed in detail. Proposed M-IHS techniques are obtained by incorporating the Heavy Ball Acceleration into the Iterative Hessian Sketch algorithm and they provide significant improvements over the randomized preconditioning techniques. By using approximate solvers along with the iterations, the proposed techniques are capable of avoiding all matrix decompositions and inversions, which is one of the main advantages over the alternative solvers such as the Blendenpik and the LSRN. Similar to the Chebyshev Semi-iterations, the M-IHS variants do not use any inner products and eliminate the corresponding synchronization steps in hierarchical or distributed memory systems, yet the M-IHS converges faster than the Chebyshev Semi-iteration based solvers. Lower bounds on the required sketch size for various randomized distributions are established through the error analyses of the M-IHS variants. Unlike the previously proposed approaches to produce a solution approximation, the proposed M-IHS techniques can use sketch sizes that are proportional to the statistical dimension which is always smaller than the rank of the coefficient matrix. Additionally, hybrid schemes are introduced to estimate the unknown $\ell 2$-norm regularization parameter along with the iterations of the M-IHS techniques. Unlike conventional hybrid methods, the proposed Hybrid M-IHS techniques estimate the regularization parameter from the lower dimensional sub-problems that are constructed by random projections rather than the deterministic projections onto the Krylov Subspaces. Since the lower dimensional sub-problems that arise during the iterations of the Hybrid M-IHS variants are close approximations to the Newton sub-systems and the accuracy of their solutions increase exponentially, the parameters estimated from them rapidly converge to a proper regularization parameter for the full problem.

In various numerical experiments conducted at several noise levels, the `Hybrid M-IHS` variants consistently estimated better regularization parameters and constructed solutions with less errors than the direct methods in far fewer iterations than the conventional hybrid methods. In large scale applications where the coefficient matrix is distributed over a memory array, the proposed `Hybrid M-IHS` variants provide improved efficiency by minimizing the number of distributed matrix-vector multiplications with the coefficient matrix.

Keywords: Least squares, Tikhonov regularization, ridge regression, random projection, oblivious subspace embeddings, randomized preconditioning, acceleration, hybrid methods.

ÖZET

BÜYÜK ÖLÇEKLİ DOĞRUSAL EN KÜÇÜK KARELER PROBLEMLERİ İÇİN HIZLI VE GÜRBÜZ ÇÖZÜM YÖNTEMLERİ

İbrahim Kurban Özaslan
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Orhan Arıkan
Temmuz 2020

Büyük ölçekli ve doğrusal en küçük kareler problemleri için bir grup çözücü olan Momentum Yinelemeli Hessian Krokileme (`M-IHS`) teknikleri önerilmiş ve analiz edilmiştir. Önerilen `M-IHS` teknikleri, Ağır Top Hızlandırmasının Yinelemeli Hessian Krokileme algoritmasına dahil edilmesiyle elde edilir ve rastlantısal ön koşullandırma teknikleri üzerinde önemli gelişmeler sağlar. Önerilen teknikler, yinelemelerle birlikte yaklaşık çözücüler kullanarak tüm matris ayrışmalarından ve ters çevirmelerden kaçınabilir, bu nedenle önerilen yöntemler büyük ölçekli problemlerde Blendenpik ve LSRN gibi alternatif çözücülere göre daha avantajlıdır. Chebyshev Yarı-iterasyonlarına benzer şekilde, `M-IHS` varyantları da yinelemeler sırasında herhangi bir iç çarpım kullanmaz, dolayısıyla hiyerarşik veya dağıtılmış bellek sistemlerinde iç çarpım hesaplamalarının neden olduğu senkronizasyon adımlarını ortadan kaldırır ve önerilen `M-IHS` teknikleri Chebyshev Yarı-iterasyonlarına dayalı çözümlerden daha hızlı bir şekilde çözüme yakınsar. Çeşitli rasgele dağılımlar için gerekli olan en küçük çizim boyutu, önerilen tekniklerin hata analizleri yoluyla belirlenmiştir. Önerilen `M-IHS` teknikleri çözüm yaklaşıklaması üretmek için, daha önce önerilen yaklaşımların aksine, katsayı matrisinin kertesinden her zaman daha küçük olan istatistiksel boyutla orantılı bir kroki matris boyutu kullanabilir. Tüm bunlara ek olarak, ℓ2-norm düzenlileştirme parametresinin bilinmediği durumlarda, bu parametreyi M-IHS tekniklerinin yinelemeleri sırasında tahmin etmek için melez şemalar önerilmiştir. Önerilen Melez `M-IHS` şemaları düzenlileştirme parametresini, gerekirci projeksiyonlar yoluyla elde ettiği Krylov Altuzayları'nı kullanarak tahmin eden geleneksel melez yöntemlerden farklı olarak, rastgele projeksiyonlarla oluşturduğu daha düşük boyutlu alt problemlerden tahmin eder. Melez `M-IHS` yinelemeleri sırasında ortaya çıkan bu düşük boyutlu alt problemler, Newton alt sistemlerine yakın yaklaşıklamalar olduğundan ve bu alt problemlerin çözümlerinin doğruluğu katlanarak arttığından, bu alt problemlerden tahmin edilen düzenlileştirme parametreleri hızla tam problem kullanılarak tahmin edilen parametrelere yakınsar. Farklı gürültü seviyelerinde yapılan çeşitli sayısal deneylerde, Melez M-IHS şemaları, doğrudan yöntemler aracılığıyla tam problemden tahmin edilen düzenlileştirme parametrelerinden daha az hataya sebep olan parametreleri ve bu parametrelere denk gelen çözümleri geleneksel melez yöntemlerden çok daha az yineleme gerektirerek üretmiştir. Katsayı matrisinin bir bellek dizisi üzerinde dağıtıldığı büyük ölçekli uygulamalarda, önerilen Melez M-IHS şemaları katsayı matrisi kullanılarak hesaplanan dağıtılmış matris-vektör çarpımlarının sayısını en aza indirerek önemli bir verimlilik sağlamaktadır.

Anahtar sözcükler: En küçük kareler, Tikhonov düzenlileştirmesi, rastlanstısal projeksiyon, rastgeleleştirilmiş ön şartlandırma, hızlandırma, melez metotlar.

Acknowledgement

I would like to give my foremost thanks to my advisor Orhan Arıkan for guiding me through endless difficulties of graduate school and for encouraging me to freely explore my interests. He taught me not only how to do first-class research, but also the fundamentals of academic writing, which is reflected in every corner of this thesis. I also thank him for his availability even amid holidays and for his patience with my scrawls on the whiteboard during our meetings.

I was extremely fortunate to work with Mert Pilancı. He has been a great inspiration throughout my studies and has a remarkable influence on my entire research. I am greatly indebted to him for giving ideas that spike the foundations of the work in this thesis.

I would also like to thank Sinan Gezici and Elif Vural for being in my thesis committee and for their valuable comments. I also acknowledge the financial support of The Scientific and Technological Research Council of Turkey (TÜBİTAK) under the 2210-A program during my graduate study.

My entire experience in grad school, especially long study hours and stressful moments, have become more bearable thanks to the friendship of Talha Akyıldız. Also, I wish to thank my office mate Ertan Kazıklı. He has always been very kind and helpful whenever I need.

My heartfelt gratitude goes to my parents Müşerref and Mustafa for their continuous caring throughout my life. They always believe in me even at times I do not believe in myself.

Finally, I want to express my deepest appreciation to my better half Kübra Keskin for her company, support, and wisdom. Without her, my life would be dark and gloomy. Her presence brought a colorful light into my life.

Contents

List of Figures

List of Tables

List of Abbreviations

A-IDRP	Accelerated Iterative Dual Random Projection
A-IHS	Accelerated Iterative Hessian Sketch
A-IPDS	Accelerated Iterative Primal Dual Sketch
ADMM	Alternating Direction Method of Multipliers
AMM	Approximate Matrix Property
ARK	Accelerated Randomized Kaczmarz
CG	Conjugate Gradient
CS	Chebyshev Semi-iterative
DCT	Discrete Cosine Transform
DP	Discrepancy Principle
GCV	Generalized Cross Validation
GKL	Golub-Kahan-Lanczos
GMRES	Generalized Minimal Residual Method
GSURE	Generalized Stein's Unbiased Risk Estimate
HS	Hessian Sketch
IHS	Iterative Hessian Sketch
LC	L-Curve

LS	Least Squares
M-IHS	Momentum Iterative Hessian Sketch
MPL	Marchenko Pastur Law
OR-LS	Oracle Regularized Least Squares
OSE	Oblivious Subspace Embedding
OSNAP	Oblivious Sparse Norm-Approximating Projections
ROS	Randomized Orthogonal System
RP	Random Projection
SNR	Signal-to-Noise Ratio
SRHT	Subsampled Randomized Hadamard Transform
SVD	Singular Value Decomposition
TSVD	Truncated Singular Value Decomposition
UPRE	Unbiased Predictive Risk Estimate
WGCV	Weighted Generalized Cross Validation

Chapter 1

Introduction

The foundation of linear inverse problems goes back to Babylonia, an ancient Mesopotamian kingdom around 4000 years ago, where people know how to solve 2×2 dimensional linear system of equations [2]. Chinese mathematicians around 200 C.E. had developed techniques to solve $n \times n$ dimensional linear systems of equations that are very similar to Gaussian Elimination [3]. Developments in solving the linear equations did not come to fruition until the invention of the method of Least Squares (LS) around the late 18^{th} century. After that, the techniques to find or to approximate a solution of a linear system that has either a unique solution or infinite number of solutions or even that does not have any solution have attracted more and more attention from diverse fields of science and engineering [4]. By the advancements in digital computing around 1950s, many algorithms developed until that time were recognized to perform poorly on well conditioned problems due to the finite precision used to represent real numbers in the computers. Then, importance of the direct methods that solve the linear systems by factorizing the coefficient matrices into simpler factors started to grow [5]; but at the same time, memory limitations in the computing devices make iterative methods such as the Krylov Subspace techniques attractive means for approximating the solutions [6].

The LS solution of an $n \times d$ dimensional linear system can be found in $O(nd \min(n, d))$ operations by computing Cholesky factorization or QR decomposition of the coefficient matrix [7], but due to the quadratic dependence on the dimensions, cost of the solution becomes excessively high for large scale problems. Linear dependence on the dimensions might be seen as acceptable for large scale problems and can be realizable by using the first order iterative methods that are based on the Krylov Subspaces [8]. Starting with an approximate solution, these methods increase its accuracy iteratively where at each iteration $O(nd)$ operations are computed. However, the number of iterations to obtain an accurate solution can be extremely large for ill-posed problems. Preconditioning techniques that aim to map the problem to a well conditioned one can be used to reduce the number of iterations, but unless the coefficient matrix has a special structure, finding a low cost and effective preconditioning matrix is still a challenging problem [9].

When the linear system is corrupted by the measurement noise or discretization errors, the LS methods produces unacceptably noisy reconstructions in ill-posed problems. To provide robust solutions, an $\ell 2$-norm penalty on the magnitude of the solution is introduced to the formulation. Finding a proper regularization parameter is another issue in the iterative solvers, which affects the number of iterations and the error of the regularized solution. There are techniques which estimate the regularization parameter along with the iterations. For example, techniques such as the LSQR and the GMRES explicitly construct an orthogonal basis for the Krylov subspace through iterations and thus allow estimation of the regularization parameter in a lower dimensional subspace [10]. In severely ill-conditioned problems, a suitable regularization parameter is typically estimated in a few iterations, therefore these techniques, that are referred to as the hybrid methods, do not face severe complexity limitations. However, severely ill-conditioned problems form a small subset of linear inverse problems in practice (see for example distribution of the singular value profiles in [11]) and the number of iterations required to estimate a robust regularization parameter through the conventional hybrid methods can grow unpredictably large for a milder level of ill-conditioning.

In addition to the total operation count, there are two decisive factors for determining the feasibility of the algorithms in large scale problems. The first one is the number of matrix-vector multiplications with the coefficient matrix. In computational environments where the coefficient matrix is stored in a memory network, distributed computation of the matrix-vector multiplications causes prohibitively long run times [12]. The second factor is the number of inner product calculations in the iterations. Each inner product calculation corresponds to a synchronization step in parallel computing and causes high communication costs in distributed or hierarchical memory systems [13].

The aforementioned drawbacks of the deterministic methods can be remedied by using the Random Projection (RP) techniques [14]. These techniques are capable of both reducing the dimensions and bounding the number of iterations with statistical guarantees, while they are quite convenient for parallel and distributed computations. The development and the applications of the RP based algorithms can be found in [15, 16, 17] and references therein.

In this thesis, a family of RP-based iterative solvers is proposed for large scale linear LS problems. The asymptotic and non-asymptotic analyses show that the iterations of the proposed solvers converge to the optimal solution of the LS problem at an exponential rate which is independent of the spectral properties of the coefficient matrix. Therefore, the number of iterations for the proposed solvers to reach any level of accuracy is bounded. The proposed solvers require only one matrix-vector multiplication per iteration with the coefficient matrix and do not require any inner product calculations. Hence, they are efficient not only in the sequential computing systems but also in parallel and distributed memory environments. In the absence of the regularization parameters, hybrid schemes are introduced for the proposed solvers. The proposed hybrid schemes, that are based on random projections rather than the deterministic projections onto the Krylov Subspaces, do not increase the number of accesses to the coefficient matrix and find better regularization parameters in far fewer iterations than the conventional hybrid methods.

1.1 Organization of the Thesis

Chapter 2 begins with the formulation of the LS problems, then gives an overview of the existing deterministic approaches to find the solution and to estimate the regularization parameter. Afterwards it reviews the applications of the RP-based methods to the LS problems.

Chapter 3 assumes that a proper estimate of the regularization parameter is available for the regularized LS problems, and introduces the proposed `M-IHS` variants. Then, convergence analyses of the proposed solvers in both asymptotic and non-asymptotic dimension regimes are given. A stable and efficient solver that is particularly designed for the sub-problems that arise during the iterations of all the `M-IHS` variants is also proposed in this chapter.

Chapter 4 focuses on the estimation of the regularization parameter. First, the bottleneck in the conventional hybrid methods are discussed, and then the RP-based hybrid schemes for the `M-IHS` variants are derived. Finally, Chapter 5 summarizes the findings of this thesis and discusses the future work.

1.2 Notation

Throughout the thesis, bold letters are used for vectors and matrices such as $\mathbf{x}, \mathbf{b}$ and $\mathbf{A}, \mathbf{S}$. Euclidean norm is stated as $\|\cdot\|_2$, the weighted Euclidian norm is shown as $\|\mathbf{x}\|_{\mathbf{W}} = \|\mathbf{W}\mathbf{x}\|_2$ and $\mathtt{tr}\left(\cdot\right)$ denotes the trace of the argument. The superscript $\mathbf{A}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$.

Chapter 2

Review of Available Techniques for Linear Least Squares Problems

This chapter starts with the formulation of the LS problems, then reviews existing deterministic approaches that are used for both construction of the solution and estimation of the regularization parameter. The RP-based approaches used for the LS problems are also discussed in this chapter.

2.1 Linear Least Squares Problems

The thesis focuses on fast and robust solution techniques for large scale linear systems of equations in the form of

$$\mathbf{A}\mathbf{x}_0 + \mathbf{w} = \mathbf{b} \tag{2.1}$$

where $\mathbf{A}$ is the given data or coefficient matrix, which might be an operator as well, and $\mathbf{b}$ is the given measurement or observation vector. The entries of $\mathbf{b}$

are contaminated by the measurement noise or computation/discretization errors represented as $\mathbf{w}$. The aim of the linear inverse problems is to obtain an accurate estimate for the true input $\mathbf{x}_0$ by observing $(\mathbf{A}, \mathbf{b})$ pair. In this thesis we are particularly interested in the solutions that minimize the mean squared error, referred to as the *Least Squares* (LS) solution:

$$\mathbf{x}_{\mathrm{LS}} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^d} \underbrace{\frac{1}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{b}\right\|_2^2}_{f(\mathbf{x})}. \tag{2.2}$$

In practice, due to the commonly encountered ill conditioned nature of $\mathbf{A}$, the quadratic objective function in eq. (2.2) may not produce acceptable results. Therefore it is generally used with an additional penalty term on the magnitude of the solution as:

$$\mathbf{x}(\lambda) = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_d)^{-1}\mathbf{A}^T\mathbf{b} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^d} \underbrace{\frac{1}{2}\left\|\mathbf{A}\mathbf{x}-\mathbf{b}\right\|_2^2 + \frac{\lambda}{2}\left\|\mathbf{x}\right\|_2^2}_{f(\mathbf{x},\lambda)}, \tag{2.3}$$

which is known as the *Tikhonov Regularization* in applied linear algebra or the *Ridge Regression* in statistics [18]. Both problems in eq. (2.2) and eq. (2.3) frequently arises in various applications of science and engineering. For example, they can appear in the discretization of Fredholm Integral Equations of the first kind [19]. In those cases, the data matrix might be ill conditioned and the linear system might be either *over-determined*, i.e., $n \geq d$, or square. When the system is *under-determined*, i.e., $n < d$, although sparse solutions are more popular due to relatively recent developments in the compressed sensing literature [20], the least norm solutions have also a considerable importance in machine learning applications such as the Support Vector Machines [21, 22]. Solutions to the problem in both dimension regimes, i.e, $n \geq d$ and $n < d$, are often required as intermediate steps of rather complicated algorithms such as the Interior Point and the ADMM that are widely used in machine learning and image processing applications [23, 24, 25].

2.2 Deterministic Approaches Used for the LS Problems

In this, first a brief review of the existing techniques to find the solution in the existence of a proper estimate of λ is presented. Then, an overview of the spectral filtering approaches and the existing methods that are used for estimation of λ is given.

2.2.1 Reconstruction for a given regularization parameter

If a proper estimate of the regularization parameter λ is available, the solution $\mathbf{x}(\lambda)$ (or $\mathbf{x}_{\text{LS}}$) can be obtained by using the *direct* methods, which are based on a variety of full matrix decomposition, rather than computing the matrix-matrix multiplication and the inversion in eq. (2.3) [7]. To use an orthogonal decomposition produces more robust solutions against rank deficiencies. However, $O(nd\min(n,d))$ computational complexity of the full matrix decompositions or $n \times d$ dimensional matrix-matrix multiplications becomes prohibitively high as the dimensions n and d increase.

For large scale problems, linear dependence on both dimensions might seem acceptable and can be realizable by using the first order *iterative* solvers that are based on the Krylov Subspaces [18]. These methods require only a few matrix-vector and vector-vector multiplications at each iteration, but the number of iterations that is needed to reach a certain level of tolerance is highly sensitive to the spectral properties of the coefficient matrix. The convergence rate of these iterative solvers including Conjugate Gradient (CG), LSQR, LSMR, Chebyshev Semi-iterative (CS) technique, GMRES and many others [8, 13] is characterized by the following inequality:

$$\left\|\mathbf{x}^i - \mathbf{x}(\lambda)\right\|_2 \leq \left(\frac{\sqrt{\kappa(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_d)} - 1}{\sqrt{\kappa(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_d)} + 1}\right)^i \left\|\mathbf{x}^1 - \mathbf{x}(\lambda)\right\|_2, \ 1 < i,$$

where $\mathbf{x}^1$ is the initial guess, $\mathbf{x}^i$ is the i^{th} iterate of the solver and the condition number $\kappa(\cdot)$ is defined as the ratio of the largest singular value to the smallest singular value of its argument [26]. Since for ill conditioned matrices $\kappa(\mathbf{A}^T\mathbf{A}+\lambda\mathbf{I}_d)$ becomes large, the rate of convergence might be extremely slow.

The computational complexity of the Krylov Subspace-based iterative solvers is $O(nd)$ for each iteration, which is significantly less than $O(nd\min(n,d))$ if the number of iterations can be significantly fewer than $\min(n,d)$. However, in applications such as big data where $\mathbf{A}$ is very large dimensional, the computational complexity is not the only metric for feasibility of the algorithms. For instance, if the coefficient matrix is too large to fit in a single working memory and it could be merely stored in a number of distributed computational nodes, then at least one distributed computations of matrix-vector multiplications are required at each iteration of algorithms such as the CGLS or the LSQR [27, 28]. Therefore the number of iterations should also be counted as an important metric to measure the overall complexity of an algorithm. One way to reduce the number of iterations in the iterative solvers is to use preconditioning to transform an ill conditioned problem to a well conditioned one with lower condition number [13]. The preconditioning can be applied to the LS problems in eq. (2.2) in two ways:

$$\begin{aligned}\text{Left Preconditioning: } \mathbf{x}_{left} &= \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \quad \left\|\mathbf{N}^T\mathbf{A}\mathbf{x}-\mathbf{N}^T\mathbf{b}\right\|_2^2,\\ \text{Right Preconditioning: } \mathbf{x}_{right} &= \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \quad \left\|\mathbf{A}\mathbf{N}\mathbf{x}-\mathbf{b}\right\|_2^2,\end{aligned}$$

where $\mathbf{x}_{left}$ and $\mathbf{N}\mathbf{x}_{right}$ equal to $\mathbf{x}_{\text{LS}}$ if and only if the range space of $\mathbf{N}\mathbf{N}^T\mathbf{A}$ is equal to the range space of $\mathbf{A}$ and $\mathbf{A}^T$, respectively [29]. In the deterministic settings, finding a low-cost and effective preconditioning matrix $\mathbf{N}$ such that $\kappa(\mathbf{A}\mathbf{N}) \ll \kappa(\mathbf{A})$ or $\kappa(\mathbf{N}^T\mathbf{A}) \ll \kappa(\mathbf{A})$ is still a challenging task unless $\mathbf{A}$ has a special structure such as being diagonally-dominant or being a band matrix [9]. Preconditioning can be applied to the regularized LS problems in eq. (2.3) in the same way as the un-regularized one by using the following formulation:

$$\mathbf{x}(\lambda) = \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \quad \left\|\mathbf{A}\mathbf{x}-\mathbf{b}\right\|_2^2 + \lambda\left\|\mathbf{x}\right\|_2^2 = \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \quad \left\|\begin{bmatrix}\mathbf{A}\\ \sqrt{\lambda}\mathbf{I}_d\end{bmatrix}\mathbf{x}-\begin{bmatrix}\mathbf{b}\\ \mathbf{0}\end{bmatrix}\right\|_2^2.$$

In addition to the number of iterations, the number of inner products in each iteration also plays an important role in the overall complexity. Each inner product calculation constitutes a synchronization step in parallel computing and therefore is undesirable for distributed or hierarchical memory systems [13]. The CS technique can be preferred in this kind of applications, since it does not use any inner products and therefore eliminates some of the synchronization steps that are required by the techniques such as the CG or the GMRES. However, the CS requires prior information about the ellipsoid that contains all the eigenvalues of $\mathbf{A}$, which is typically not available in practice [30].

2.2.2 Methods for estimation of the regularization parameter λ

We start with the review of spectral filtering approach that constitutes the foundation of the methods used for the estimation of λ. Let the Singular Value Decomposition (SVD) of $\mathbf{A}$ be $\sum\limits_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$, where $r = \min(n, d)$, $\sigma_1 \geq \ldots \geq \sigma_r \geq 0$ are the singular values, $\mathbf{u}_i$'s and $\mathbf{v}_i$'s are the left and the right singular vectors, respectively. Then, the LS solution in eq. (2.2) can be expressed as

$$\mathbf{x}_{\text{LS}} = \sum_{i=1}^{r} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i = \sum_{i=1}^{r} \left(\overline{x}_i \mathbf{v}_i + \frac{\overline{w}_i}{\sigma_i} \mathbf{v}_i \right),$$

where $|\overline{x}_i|^2 = |\mathbf{v}_i^T \mathbf{x}_0|^2$ and $|\overline{w}_i|^2 = |\mathbf{u}_i^T \mathbf{w}|^2$ represent the spectral energy of the input and the noise. If the problem is ill conditioned, the noise terms, that are amplified by the small singular values, have the dominant contribution in $\mathbf{x}_{\text{LS}}$ resulting in unacceptably noisy reconstructions for $\mathbf{x}_0$[1]. Filtering coefficients, ϕ_i's for $1 \leq i \leq r$, can be incorporated into the spectral terms in the summation to

[1] Ill posedness of the problem is not only dependent on the singular values of $\mathbf{A}$, but also dependent on the input and the noise energy distributions over the singular vectors. For the problems constructed by discretization of continuous kernels, ill posedness can be related to the *Picard Condition* [31, 32].

control the noise in the reconstruction:

$$\mathbf{x}(\mathbf{\Phi}) = \mathbf{V}\mathbf{\Phi}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{b} = \sum_{i=1}^{r} \left(\phi_i \overline{x}_i \mathbf{v}_i + \phi_i \frac{\overline{w}_i}{\sigma_i} \mathbf{v}_i \right), \tag{2.4}$$

where $\mathbf{\Phi} = \mathbf{diag}(\phi_i), 1 \leq i \leq r$ [33]. In the existence of priors on the spectral energy distributions, the filtering coefficients ϕ_i's can be selected to minimize the expected value of the *oracle error*

$$\mathbf{\Phi}^* = \underset{\mathbf{\Phi} \in \mathbb{R}^r}{\operatorname{argmin}} \ \mathbb{E}_{\mathbf{w}} \left[\|\mathbf{x}_0 - \mathbf{x}(\mathbf{\Phi})\|_2^2 \right] = \sum_{i=1}^{r} \left((1 - \phi_i)^2 \, \overline{x}_i^2 + \phi_i^2 \left(\frac{\sigma_{\mathbf{w}}}{\sigma_i} \right)^2 \right),$$

where $\mathbf{w}$ is modeled as an independent and identically distributed (i.i.d.) random vector with zero mean and covariance $\sigma_{\mathbf{w}}^2 \mathbf{I}$. Under the i.i.d. noise assumption, the optimal filtering coefficients ϕ_i^*'s in terms of the mean squared error can be found by minimizing each term in the summation as

$$\phi_i^* = \frac{\overline{x}_i^2}{\overline{x}_i^2 + \left(\frac{\sigma_{\mathbf{w}}}{\sigma_i} \right)^2} = \frac{\sigma_i^2}{\sigma_i^2 + \frac{\sigma_{\mathbf{w}}^2}{\overline{x}_i^2}}, \quad 1 \leq i \leq r,$$

which is widely known as the *Wiener Filter* in a broader setting by the signal processing community [34]. The Tikhonov regularization in eq. (2.3) corresponds to using the coefficients $\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}$ in eq. (2.4). If the input spectrum is uniformly distributed over the right singular vectors, i.e., $\mathbb{E}\left[\overline{x}_i^2\right] = \sigma_{\mathbf{x}}^2$ for $i = 1, \ldots, r$, then the regularized LS solution $\mathbf{x}(\lambda^\diamond)$ with $\lambda^\diamond = \frac{\sigma_{\mathbf{w}}^2}{\sigma_{\mathbf{x}}^2}$ will achieve the minimum mean squared error. Uniform distribution of the input spectrum might be a viable model for a set of problems in machine learning or statistics if, for example, the input signal $\mathbf{x}_0$ and $\mathbf{A}$ are drawn from independent distributions. However, data obtained from sensors typically have a decaying spectral energy distribution, since the sensors are designed to have their singular vectors associated with larger singular values aligned with the spectrum of $\mathbf{x}_0$, the desired modality of the sensing. In such cases, an apparent approximation strategy for minimizing the mean squared error is to directly exclude the degenerate input terms that have lower energy than the amplified noise, i.e., setting $\phi_i = 1$ whenever $\overline{x}_i \geq \sigma_{\mathbf{w}}/\sigma_i$

otherwise $\phi_i = 0$, which corresponds to the Truncated SVD (TSVD) solution [33]:

$$\mathbf{x}(k^*) = \sum_{i=1}^{k^*} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i \text{ with } \overline{x}_i \geq \frac{\sigma_{\mathbf{w}}}{\sigma_i} \text{ for } i \leq k^* \text{ and } \phi_i = \begin{cases} 1, & i = 1, \ldots, k^* \\ 0, & \text{otherwise} \end{cases}. \tag{2.5}$$

The truncation parameter $k^* = \sum_i^r \phi_i$ defines the *effective rank* of the problem which corresponds to the number of dimensions that the measurements are sensed more energetically than the noise. Practically, the true input information in the measurements that live outside the span of the first k^* singular vectors, i.e., $\sum_{i=k^*+1}^r \sigma_i \overline{x}_i \mathbf{u}_i$, is not recoverable due to the noise amplification phenomenon.

Instead of the TSVD, if the Tikhonov regularization is used, the above approximation can be acquired in closed form solution with a trade-off: hard thresholding of the binary coefficients in the TSVD solution is replaced by the soft thresholding of the smooth sigmoid-like filtering coefficients $\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}$. A counterpart of the *effective rank* measure is obtained by the *statistical dimension* of the problem that does not explicitly require any information about the singular values [22, 35]:

$$\mathsf{sd}_\lambda(\mathbf{A}) = \sum_{i=1}^r \phi_i = \sum_{i=1}^r \frac{\sigma_i^2}{\sigma_i^2 + \lambda} = \mathsf{tr}\left(P_{\mathbf{A}}(\lambda)\right), \tag{2.6}$$

where $P_{\mathbf{A}}(\lambda) = \mathbf{A}\left(\mathbf{A}^T\mathbf{A} + \lambda \mathbf{I}_d\right)^{-1} \mathbf{A}^T$ is a shrinkage operator called as the influence matrix [22]. The *statistical dimension* is widely used in the statistics to measure the effective degrees of freedom. As shown in Chapter 3, it determines the lower bound of the projection size for the sketched regularized LS problems [35, 36, 37].

The TSVD and the Tikhonov approaches, for properly chosen regularization parameters, are known to produce practically the same solutions if the input energy distribution is in a decaying trend, i.e., the *Discrete Picard Condition* is satisfied, or if the singular values of $\mathbf{A}$ decay at a sufficiently high rate, e.g., $\sigma_i = O(i^{-\alpha})$ for some $\alpha \geq 0$ or $\sigma_i = O(e^{-\beta i})$ for some $\beta > 1$ [38]. These rates are widely referred to as the *moderate* and the *severe* decay rates respectively in the literature [39, 40].

In the absence of the prior about spectral energy distributions, determination of the Tikhonov regularization parameter is a well studied subject for the moderate size problems. In the next section, we are going to examine some parameter selection techniques that are prevalent in practice.

2.2.2.1 Risk estimators for parameter selection

Widely used techniques can be classified under two main groups according to whether they require the noise statistics or not. The first technique that requires the prior is the Discrepancy Principle (DP) which selects λ so that the norm of the residual error becomes equal to the standard deviation of the noise:

$$\|\mathbf{A}\mathbf{x}(\lambda) - \mathbf{b}\|_2^2 = n\sigma_{\mathbf{w}}^2.$$

The DP technique is prone to overestimate the regularization parameter, which produces robust results against noise but increases the oracle error [33, 41, 42]. As an alternative, the parameter λ can be selected to minimize an unbiased risk estimator of the expected oracle error:

$$\mathbb{E}_{\mathbf{w}}\left[\|\mathbf{x}_0 - \mathbf{x}(\lambda)\|_2^2\right] = \|\mathbf{x}_0\|_2^2 + \mathbb{E}_{\mathbf{w}}\left[\|\mathbf{x}(\lambda)\|_2^2 - 2\mathbf{x}(\lambda)^T\mathbf{x}_0\right].$$

One such scheme is the Generalized Stein's Unbiased Risk Estimate (GSURE) [43], which minimizes the risk estimator $T(\lambda) = \|\mathbf{x}(\lambda)\|_2^2 - 2g(\mathbf{x}(\lambda))$ such that $\mathbb{E}_{\mathbf{w}}\left[g(\mathbf{x}(\lambda))\right] = \mathbb{E}_{\mathbf{w}}\left[\mathbf{x}(\lambda)^T\mathbf{x}_0\right]$, where different g functions for different noise distributions can be found in [42, 43]. For example, if the noise distribution is Gaussian, then the risk estimator has the following form:

$$T(\lambda) = \|\mathbf{x}_{\mathrm{LS}} - \mathbf{x}(\lambda)\|_2^2 - \sigma_{\mathbf{w}}^2 \mathsf{tr}\left((\mathbf{\Sigma}^{-2}\right) + 2\sigma_{\mathbf{w}}^2 \mathsf{tr}\left((\mathbf{\Sigma}^2 + \lambda\mathbf{I})^{-1}\right).$$

In our simulations with the Gaussian distributed noise, we found that as the noise level increases, the GSURE consistently underestimates the regularization parameter, which is also suggested by the theoretical results in [42]. In a similar but more robust approach, referred to as the Unbiased Predictive Risk Estimate

(UPRE), λ is selected to minimize an unbiased risk estimator of the *oracle predictive error*:

$$\mathbb{E}_{\mathbf{w}}\left[U(\lambda)\right] = \mathbb{E}_{\mathbf{w}}\left[\|\mathbf{A}\mathbf{x}_0 - \mathbf{A}\mathbf{x}(\lambda)\|_2^2\right] \tag{2.7}$$

where $U(\lambda) = \|\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)\|_2^2 - 2\hat{\sigma}_{\mathbf{w}}^2 \mathsf{tr}\left(\mathbf{I} - P_{\mathbf{A}}(\lambda)\right) + d\widehat{\sigma}_{\mathbf{w}}^2$ and $\widehat{\sigma}_{\mathbf{w}}^2 = \frac{1}{n-d}\|\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{LS}}\|_2^2$ [33, 42, 44].

A commonly used technique in the second group that does not require any prior information about the noise is the L-Curve (LC) technique which selects λ to maximize the curvature of the L-shaped Pareto Optimality Curve plotted $\|\mathbf{A}\mathbf{x}(\lambda) - \mathbf{b}\|_2^2$ versus $\|\mathbf{x}(\lambda)\|_2^2$ [45]. The most commonly used technique in this group is the Generalized Cross Validation (GCV) which selects λ^{gcv} as the minimizer of the the following function:

$$G_{full}(\lambda) = \frac{\|\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)\|_2^2}{\mathsf{tr}\left(\mathbf{I} - P_{\mathbf{A}}(\lambda)\right)^2}. \tag{2.8}$$

Note that the GCV is also an unbiased predictive risk estimator [44]. As $n \to \infty$, the minimizer of the expected value of the GCV and the UPRE functions converges to each other [33].

Lastly, consider the case, as we will in Chapter 4, where we can access only to the residual error that is projected onto the span of the first k left singular vectors of $\mathbf{A}$. Then, the GCV function can be modified as:

$$G_{full}(\lambda, k) = \frac{\left\|\mathbf{U}_k^T(\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda))\right\|_2^2}{\mathsf{tr}\left(\mathbf{I} - P_{\mathbf{\Sigma}_k}(\lambda)\right)^2}, \tag{2.9}$$

where $\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$ is the truncated SVD with parameter k and λ_k^{gcv} is set to the minimizer of $G_{full}(\lambda, k)$. Although exclusion of the residual parts that may contain informative statistics about the noise is expected to worsen the estimation, as long as $k > k^*$, the risk estimator $G_{full}(\lambda, k)$ produces close results to the naive GCV function $G_{full}(\lambda)$, since the measurements outside the span of the first k^* left singular vectors are already dominated by the noise and thus contain enough information to detect a separation similar to the one given in eq. (2.5). Performances of $G_{full}(\lambda, k)$ and the naive $G_{full}(\lambda)$ are compared in Section 4.3

where the comparison results of the hybrid methods are presented.

Except for the LC and the GSURE, as $n \to \infty$, expected optimal objective function value of each technique asymptotically converges to the minimizer of the expected oracle prediction error [33]. In Chapter 4, the GCV technique is preferred in the proposed hybrid methods, since it does not require any prior information and its objective function asymptotically converges to the oracle prediction error.

2.2.2.2 Conventional hybrid methods for large scale problems

Since there is no closed form solutions, the parameter selection techniques mentioned in Section 2.2.2.1 require numerical minimization to approximate the optimal regularization parameter λ. For this purpose, however, an orthogonal matrix decomposition is needed since the risk estimators depend on both $\mathbf{x}(\lambda)$ and $\mathsf{tr}\left(P_{\mathbf{A}}(\lambda)\right)$. For large scale problems, the direct use of these parameter selection techniques become infeasible due to the prohibitively high complexity of the matrix decomposition. Instead, Krylov subspace based methods can be directly applied to the linear system without any regularization [39]. Due to the semi-convergence behaviour, during the initial iterations of these solvers, the prediction error decreases until the solution information spanned by the first k^* singular vectors is completely included in the constructed Krylov subspace. Then, the prediction error starts to increase because of the inclusion of components with lower Signal to Noise Ratio (SNR) to the reconstruction. Therefore, when these solvers are terminated after a few iterations, a regularized LS solution can be obtained. For example, if $\mathbf{A}$ is a smoothing operator, the iterations can easily be terminated accurately just before the inclusion of the noise dominated components as shown in [46]; but, except for such few cases, termination of iterations requires the use of the parameter selection techniques mentioned earlier. A popular alternative approach to deal with the large scale problems is the hybrid methods [10, 40, 47, 48]. These techniques estimate a regularization parameter from the lower dimensional projected problem that is obtained in the iterations of the Golub-Kahan-Lanczos (GKL) Bidiagonalization procedure [18]. For example, at the k-th iteration of the LSQR, the following sub-problem is solved

$$\mathbf{y}^k(\lambda) = \underset{\mathbf{y}\in\mathbb{R}^k}{\operatorname{argmin}} \quad \left\|\mathbf{B}_k\mathbf{y} - \beta_1\mathbf{e}_1\right\|_2^2 + \lambda \left\|\mathbf{y}\right\|_2^2, \tag{2.10}$$

where $\mathbf{B}_k \in \mathbb{R}^{k+1\times k}$ is the lower bidiagonal matrix constructed at the k-th iteration of the GKL procedure initialized with $\mathbf{b}$ and $\beta_1 = \left\|\mathbf{b}\right\|_2$, and $\mathbf{e}_1$ is the first canonical basis vector [10, 49]. The parameter λ can be estimated from this lower dimensional sub-problem by modifying the GCV as

$$G_{proj}(\lambda) = \frac{\left\|\beta_1\mathbf{e}_1 - \mathbf{B}_k\mathbf{y}^k(\lambda)\right\|_2^2}{\mathsf{tr}\left(\mathbf{I}_{k+1} - P_{\mathbf{B}_k}(\lambda)\right)^2}. \tag{2.11}$$

Reorthogonalization steps for the GKL procedure are necessary to have the above estimator, even though the iterative solvers used in the hybrid updates do not require reorthogonalization [40, 46, 50]. As long as the size of the bidiagonal matrix, k, is sufficiently small, the GCV function in eq. (2.11) can be minimized by computing the SVD of the $\mathbf{B}_k$ in practice [10, 40]. The estimator G_{proj} is known to overestimate the regularization parameter of the full problem. To avoid overestimation, Weighted GCV(W-GCV) technique has been proposed in [51]:

$$G_{proj}(\lambda, \omega) = \frac{\left\|\beta_1\mathbf{e}_1 - \mathbf{B}_k\mathbf{y}^k(\lambda)\right\|_2^2}{\mathsf{tr}\left(\mathbf{I}_{k+1} - \omega P_{\mathbf{B}_k}(\lambda)\right)^2}, \quad \omega \in [0,\ 1] \tag{2.12}$$

with a heuristic algorithm to estimate ω. Although authors in [51] discuss some possibilities, they did not indicated the main reason behind the overestimation problem of the naive GCV technique applied on the bidiagonal system. We observe that the actual reason behind the overestimation is the use of erroneously determined degrees of freedom of the residual error as discussed in Section 4.1.

2.3 Random Projection Based Approaches for the Solutions of LS Problems

The randomness in these techniques is based on a particular mechanism used for the dimension reduction that will be detailed in this section.

Definition 1. (Oblivious $\ell 2$ Subspace Embedding (OSE) [52]) *If a distribution $\mathcal{D}$ over $\mathbb{R}^{m\times n}$ satisfies the following concentration inequality*

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}}\left(\left\|\mathbf{U}^T\mathbf{S}^T\mathbf{S}\mathbf{U}-\mathbf{I}\right\|_2 > \epsilon\right) < \delta,$$

with $\forall\mathbf{U}\in\mathbb{R}^{n\times k},\ \mathbf{U}^T\mathbf{U}=\mathbf{I}_k,\ \mathbf{S}\in\mathbb{R}^{m\times n}$*, then it is called* (ϵ,δ,k)*-OSE.*

The *sketching* matrix $\mathbf{S}$ sampled from the distribution $\mathcal{D}$ transforms any set points in high dimensional subspace, i.e., range space of $\mathbf{U}$, to a lower dimensional space $\mathbb{R}^m$ in such a way that the distance between the points in the set are nearly preserved with certain probability. Such embeddings are simple to obtain and to apply, at the same time, extremely powerful techniques to reduce the dimensions and to gain significant computational savings. For example, if the entries of $\mathbf{S}$ is drawn from the normal distribution $\mathcal{N}(0,1/m)$ then the *sketch size* m can be chosen proportional to $\epsilon^{-2}\log(1/\delta)$ in order to obtain a (ϵ,δ,n)-OSE, i.e., $\mathbf{S}$ is capable of transforming any set of point in the entire $\mathbb{R}^n$ to $\mathbb{R}^m$ in the sense given in Definition 1 [53].

In the existence of a proper estimate of the regularization parameter λ, there are two main approaches for the applications of the OSEs to the LS problems in eq. (2.2) and eq. (2.3). In the first approach, that is referred to as the *classical sketching*, the coefficient matrix $\mathbf{A}$ and the measurement vector $\mathbf{b}$ are projected down onto a lower dimensional subspace by using a randomly constructed sketching matrix $\mathbf{S}\in\mathbb{R}^{m\times n}$ with $m \ll n$, to obtain efficiently an ζ-optimal solution with high probability for the *cost approximation* [14, 54]:

$$\widetilde{\mathbf{x}} = \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}}\ \frac{1}{2}\left\|\mathbf{S}\mathbf{A}\mathbf{x}-\mathbf{S}\mathbf{b}\right\|_2^2 + \frac{\lambda}{2}\left\|\mathbf{x}\right\|_2^2,\ \text{s.t.}\ f(\widetilde{\mathbf{x}},\lambda) \le (1+\zeta)f(\mathbf{x}(\lambda),\lambda). \quad (2.13)$$

For both the sparse and dense systems, in [36] the best known lower bounds on the sketch size for obtaining an ζ-optimal cost approximation have been derived showing that the sketch size can be chosen proportional to the statistical dimension which is defined in eq. (2.6). Although the cost approximation is sufficient for many machine learning problems, the *solution approximation* which aims to produce solutions that are close to the optimal solution is a more preferable metric for the typical inverse problems [18, 33]. However, as shown in [55], the classical sketching is sub-optimal in terms of the minimum sketch size for obtaining a solution approximation.

In the second approach of *randomized preconditioning*, by iteratively solving a number of low dimensional sub-problems constituted by $(\mathbf{SA}, \nabla f(\mathbf{x}^i, \lambda))$ pairs, algorithms with reasonable sketch sizes obtain an η-optimal solution approximation:

$$\|\widehat{\mathbf{x}} - \mathbf{x}(\lambda)\|_{\mathbf{X}} \leq \eta \, \|\mathbf{x}(\lambda)\|_{\mathbf{X}} \, , \tag{2.14}$$

where $\widehat{\mathbf{x}}$ is the solution estimate and $\mathbf{X}$ is a positive definite weight matrix. In [56], OSEs have been utilized to construct a preconditioning matrix for CG-like algorithms. For this purpose, the inverse of $\mathbf{R}$-factor in the QR decomposition of the *sketched matrix* $\mathbf{SA}$ has been used as a preconditioning matrix. Later, implementation of similar ideas resulted in Blendenpik and LSRN which have been shown to be faster than some of the deterministic solvers of LAPACK [29, 57]. To solve the preconditioned problems, as opposed to the Blendenpik which uses the LSQR, the LSRN uses the CS technique for parallelization purposes and deduce the prior information about the eigenvalues based on the results of the random matrix theory. The main drawback of the LSRN and the Blendenpik is that regardless of the desired accuracy η, one has to pay the whole cost, $O(md^2)$, of a full $m \times d$ dimensional matrix decomposition, which is the dominant term in the computational complexity of these algorithms. Moreover, in the large scale inverse problems such as 3D imaging [58], even the decomposition of $m \times d$-dimensional sketched matrix may not be feasible. Iterative Hessian Sketch (IHS) [55], which is originally designed for solving constrained convex problems, follows a somewhat different path than the approach proposed in [56] and approximates the Hessian of the quadratic objective given in eq. (2.2) with the sketched matrices to gain computational saving while calculating the matrix-matrix multiplication. The objective function given in eq. (2.2) can be formulated as a combination of the Hessian and the Jacobian term:

$$\mathbf{x}_{\text{LS}} = \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \ \frac{1}{2}\left\|\mathbf{A}(\mathbf{x}-\mathbf{x}^0)\right\|_2^2 - \langle \mathbf{A}^T(\mathbf{b}-\mathbf{A}\mathbf{x}^0),\ \mathbf{x}\rangle, \tag{2.15}$$

where x^0 is any initial vector. The IHS approximates the Hessian term and uses the following updates to increases the accuracy of the iterations:

$$\begin{aligned}\mathbf{x}^{i+1} &= \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \ \frac{1}{2}\left\|\mathbf{S}_i\mathbf{A}(\mathbf{x}-\mathbf{x}^i)\right\|_2^2 - \langle \mathbf{A}^T(\mathbf{b}-\mathbf{A}\mathbf{x}^i),\ \mathbf{x}\rangle \\ &= \mathbf{x}^i + \left(\mathbf{A}^T\mathbf{S}_i^T\mathbf{S}_i\mathbf{A}\right)^{-1}\mathbf{A}^T\left(\mathbf{b}-\mathbf{A}\mathbf{x}^i\right). \end{aligned} \tag{2.16}$$

where $\mathbf{S}_i \in \mathbb{R}^{m\times n}$ with $m \ll n$ and $\mathbb{E}\left[\mathbf{S}_i^T\mathbf{S}_i\right] = \mathbf{I}_n$. In [17], it is suggested that the sketching matrix $\mathbf{S}_i = \mathbf{S}$ can be generated once and be used for all iterations in unconstrained problems. In that case, the updates in eq. (2.16) can be interpreted as the Preconditioned Gradient Descent Method that uses the sketched Hessian $\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}$ as the preconditioning matrix [1]. However, for small sketch sizes, using the same sketching matrix might cause iterations to diverge from the solutions [1]. To prevent this divergent behaviour, instead of finding a proper step size, Wang et al. used the preconditioning idea of the IHS in the CG technique and proposed the Accelerated IHS (A-IHS). Unlike gradient descent method, the CG technique does not need parameter tuning for the step size and enjoys faster convergence [59].

Chapter 3

Proposed M-IHS Techniques

This chapter consists of five main sections. In Section 3.1, the `M-IHS` technique is derived for highly over-determined un-regularized LS problems and its convergence behaviour is analyzed through the asymptotic results in the random matrix theory. Section 3.2 focuses on the $\ell2$-norm regularized LS problems formulated in eq. (2.3). In this section, the theory of the `M-IHS` is extended to the highly under-determined problems by using the convex duality and the `Dual M-IHS` technique is derived as a result. The non-asymptotic convergence analyses of the `M-IHS` and the `Dual M-IHS` are established. In the light of their convergence properties, the `Primal Dual M-IHS` techniques are introduced to reduce the dimensions of the coefficient matrix from both sides. In Section 3.3, an efficient and stable sub-solver, referred to as `AAb_Solver`, that is particularly designed for the linear systems in the form of $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{b}$ which arise during the iterations of the all `M-IHS` variants, is proposed. Numerical comparisons of the proposed `M-IHS` techniques with the state-of-the-art solvers are given in Section 3.4. The chapter is ended in Section 3.5 by stating the contributions and the conclusion remarks. Note that the findings obtained in section 3.1 has been presented in ICASSP, 2019 [60]. The rest of the analyses in the chapter is presented in the pre-print that is available in arXiv [37].

3.1 Unregularized LS Problems: Derivation and Asymptotic Analysis of the M-IHS Technique

In this section, we first show that if a fixed sketching matrix $\mathbf{S}_i = \mathbf{S}$ is used for all iterations in eq. (2.16), then the optimal step size that maximizes the convergence rate can be estimated by using the asymptotic results established for the singular value distribution of the random matrices. After then, the proposed Momentum-IHS technique is derived by extending the asymptotic analysis for the additional momentum term that accelerates the convergence.

To minimize the quadratic objective function of the unregularized LS problems given in eq. (2.2), instead of using IHS updates in eq. (2.16), we will use a single sketching matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$, such that $\mathbb{E}\left[\mathbf{S}^T\mathbf{S}\right] = \mathbf{I}_n$, for all iterations in the following damped-IHS update:

$$\mathbf{x}^{i+1} = \mathbf{x}^i + t\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\left(\mathbf{b} - \mathbf{A}\mathbf{x}^i\right), \tag{3.1}$$

where t is the fixed step size that prevents the divergent behaviour shown in [1] when fixed sketching matrix is used for all IHS iterations. In this way, we aim to reduce complexity of the IHS updates given in eq. (2.16) considerably. The following theorem states that as the dimensions of $\mathbf{A}$ go to ∞, the damped-IHS given in eq. (3.1) converges to the solution $\mathbf{x}_{\text{LS}}$ with an exponentially decaying error upper bound.

Theorem 3.1.1. *Let* $\mathbf{A}$ *and* $\mathbf{b}$ *be the given data in eq.* (2.1)*. As* n, d, m *go to* ∞ *while the ratio* $\rho = d/m$ *remains the same, if the entries of the sketching matrix* $\mathbf{S}$ *are independent, zero mean, unit variance with bounded higher order moments, then the damped-IHS updates in eq.* (3.1) *with the step size* $\frac{(1-\rho)^2}{1+\rho}$ *converges to the optimal solution* $\mathbf{x}_{LS}$ *with the following rate*

$$\left\|\mathbf{x}^i - \mathbf{x}_{LS}\right\|_{\mathbf{\Sigma}} \leq \left(\frac{2\sqrt{\rho}}{1+\rho}\right)^i \left\|\mathbf{x}^0 - \mathbf{x}_{LS}\right\|_{\mathbf{\Sigma}}, \tag{3.2}$$

where $\mathbf{\Sigma} = \mathbf{diag}(\sigma_1, \ldots, \sigma_d)$ *and* σ_i *is the* i^{th} *singular value of* $\mathbf{A}$*.*

Proof. The convergence behaviour of the damped-IHS can be investigated by finding the transformation matrix between the current and the previous error vectors through the same approach in [61]. The $\ell 2$-norm of the transformation matrix serves as a lower bound for the convergence rate. For this purpose, consider the following transformation:

$$\begin{aligned}\left\|\mathbf{x}^{i+1} - \mathbf{x}_{\mathrm{LS}}\right\|_2 &= \left\|\mathbf{x}^i + t\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\left(\mathbf{b} - \mathbf{A}\mathbf{x}^i\right) - \mathbf{x}_{\mathrm{LS}}\right\|_2 \\ &= \left\|\left(\mathbf{I}_d - t\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{A}\right)\left(\mathbf{x}^i - \mathbf{x}_{\mathrm{LS}}\right)\right\|_2 \\ &\leq \| \underbrace{\left\|\mathbf{I}_d - t\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{A}\right\|_2}_{\mathbf{T}} \left\|\mathbf{x}^i - \mathbf{x}_{\mathrm{LS}}\right\|_2\end{aligned}$$

Therefore, we can write following improvement by using the Gelfand Formula:

$$\begin{aligned}\left\|\mathbf{x}^i - \mathbf{x}_{\mathrm{LS}}\right\|_2 &\leq \left\|\mathbf{T}^i\right\|_2 \left\|\mathbf{x}^0 - \mathbf{x}_{\mathrm{LS}}\right\|_2 \\ &\leq \left(\varrho(\mathbf{T})^i + \epsilon_i\right)\left\|\mathbf{x}^0 - \mathbf{x}_{\mathrm{LS}}\right\|_2,\end{aligned} \tag{3.3}$$

where $\lim\limits_{i\to\infty} \epsilon_i = 0$ and $\varrho(\mathbf{T})$ is the spectral radius of $\mathbf{T}$. If the spectral radius is bounded, then contraction ratio (or the norm of transformation) can be bounded as well. To find $\varrho(\mathbf{T})$, the largest and the smallest eigenvalues of matrix $\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{A}$ should be determined. Changing basis by using $(\mathbf{A}^T\mathbf{A})^{-1/2}$ yields $(\mathbf{A}^T\mathbf{A})^{1/2}\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}(\mathbf{A}^T\mathbf{A})^{1/2}$ which is a symmetric matrix similar to $\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{A}$. By using compact SVD of $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, we obtain

$$\begin{aligned}\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{A} &\sim \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T(\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{S}^T\mathbf{S}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{\Sigma}\mathbf{V}^T \\ &= \mathbf{V}(\mathbf{U}^T\mathbf{S}^T\mathbf{S}\mathbf{U})^{-1}\mathbf{V}^T.\end{aligned} \tag{3.4}$$

Since $\mathbf{V}$ is a unitary matrix, spectral properties depends only on the eigenvalues of $(\mathbf{U}^T\mathbf{S}^T\mathbf{S}\mathbf{U})^{-1}$. The entries of $\mathbf{S}\mathbf{U}$ have the same probability distribution as the entries of $\mathbf{S}$ because the columns of $\mathbf{U}$ is an orthonormal set of vectors and entries of $\mathbf{S}$ are zero mean, unit variance i.i.d. random variables. Hence, if we generate a sketch matrix $\overline{\mathbf{S}} \in \mathbb{R}^{m\times d}$ with the same techniques used for $\mathbf{S}$, then

$\mathbf{SU}$ will be statistically equivalent to $\overline{\mathbf{S}}$.

Based on this observation, we need to know the largest and the smallest eigenvalues of a sample covariance matrix of $\overline{\mathbf{S}} \in \mathbb{R}^{m\times d}$ which is named as the *Wishart* matrix in statistics [62]. By the Marchenko Pastur Law (MPL), the largest and the smallest eigenvalues of the Wishart matrices bounded by the interval $[(1-\sqrt{d/m})^2,\ (1+\sqrt{d/m})^2]$, as $m \to \infty$ while the ratio d/m remains the same [63, 64]. Therefore, the largest and the smallest eigenvalues of $\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{A}$ are also asymptotically bounded by the interval $[1/(1+\sqrt{d/m})^2,\ 1/(1-\sqrt{d/m})^2]$, and the spectral radius $\varrho(\mathbf{T})$ is:

$$\varrho(\mathbf{T}) = \max\left\{\left|1-\frac{t}{\left(1+\sqrt{\rho}\right)^2}\right|, \left|1-\frac{t}{\left(1-\sqrt{\rho}\right)^2}\right|\right\}.$$

Here, the following choice for t yields the minimum spectral radius

$$t = \frac{2\cdot(1+\sqrt{\rho})^2(1-\sqrt{\rho})^2}{(1+\sqrt{\rho})^2+(1-\sqrt{\rho})^2} = \frac{(1-\rho)^2}{1+\rho},$$

which remains constant during the iterations. The resulting spectral radius is

$$\varrho(\mathbf{T}) = \left|1-\frac{(1-\rho)^2}{(1+\rho)\left(1+\sqrt{\rho}\right)^2}\right| = \frac{2\sqrt{\rho}}{1+\rho}.$$

The Gelfand formula given in eq. (3.3) concludes the proof. The weight of the norm is due to the change of the basis used in eq. (3.4) while finding the eigenvalues of $\mathbf{T}$. □

The proposed Momentum-IHS is obtained by incorporating the Heavy Ball Acceleration [65] into the damped-IHS updates given in eq. (3.1). The Heavy Ball Acceleration creates the momentum effect in the updates of Gradient Descent (GD) by taking a step along with the linear combination of the two gradients: the gradient of the objective function and the gradient of the trajectory, i.e.,

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha_i \nabla f(\mathbf{x}^i) + \beta_i(\mathbf{x}^i - \mathbf{x}^{i-1}),$$

where α_i and β_i are respective momentum weights. The `M-IHS` is obtained in the same way by adding a momentum term into to the updates of the damped-IHS as following:

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha \left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1} \mathbf{A}^T \left(\mathbf{b} - \mathbf{A}\mathbf{x}^i\right) + \beta \left(\mathbf{x}^i - \mathbf{x}^{i-1}\right) \tag{3.5}$$

where α and β are the fixed momentum parameters that are chosen to maximize the convergence rate.

Theorem 3.1.2. *Let* $\mathbf{A}$ *and* $\mathbf{b}$ *be the given data in eq.* (2.1)*. As* n, d, m *go to* ∞ *while the ratio* $\rho = d/m$ *remains the same, if the entries of the sketching matrix* $\mathbf{S}$ *are independent, zero mean, unit variance with bounded higher order moments, then the* `M-IHS` *applied on the problem given in eq.* (2.2) *with the following momentum parameters*

$$\beta = \rho, \qquad \alpha = (1-\rho)^2 \tag{3.6}$$

converges to the optimal solution $\mathbf{x}_{LS}$ *with the following rate:*

$$\left\|\mathbf{x}^{i+1} - \mathbf{x}_{LS}\right\|_{\mathbf{\Sigma}} \leq \left(\sqrt{\frac{d}{m}}\right)^i \left\|\mathbf{x}^0 - \mathbf{x}_{LS}\right\|_{\mathbf{\Sigma}}, \tag{3.7}$$

where $\mathbf{\Sigma} = \mathbf{diag}(\sigma_1, \ldots, \sigma_d)$ *and* σ_i *is the* i^{th} *singular value of* $\mathbf{A}$*.*

Proof. Consider the following bipartite transformation between two consecutive iterations of the `M-IHS`:

$$\begin{bmatrix} \mathbf{x}^{i+1} - \mathbf{x}_{\mathrm{LS}} \\ \mathbf{x}^i - \mathbf{x}_{\mathrm{LS}} \end{bmatrix} = \underbrace{\begin{bmatrix} (1+\beta)\mathbf{I}_d - \alpha \left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1} \mathbf{A}^T\mathbf{A} & -\beta\mathbf{I}_d \\ \mathbf{I}_d & \mathbf{0} \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} \mathbf{x}^i - \mathbf{x}_{\mathrm{LS}} \\ \mathbf{x}^{i-1} - \mathbf{x}_{\mathrm{LS}} \end{bmatrix}$$

By using the same similarity transformation given in [59, 61], a block diagonal form for the transformation matrix $\mathbf{T}$ can be found to determine its eigenvalues easily. For this purpose, the following change of basis will be used:

$$
\begin{aligned}
&\mathbf{T} = \mathbf{P}^{-1}\,\mathbf{diag}(\mathbf{T}_1, \ldots, \mathbf{T}_d)\mathbf{P} &\qquad& \mathbf{T}_k : = \begin{bmatrix} 1+\beta-\alpha\mu_k & \beta \\ 1 & 0 \end{bmatrix} \\
&\mathbf{P} = \begin{bmatrix} \mathbf{\Upsilon} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Upsilon} \end{bmatrix} \mathbf{\Pi}, &\qquad& \Pi_{k,\ell} = \begin{cases} 1 & \text{k odd } \ell = k, \\ 1 & \text{k even } \ell = d+k, \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.8}
$$

where $\mathbf{\Upsilon}\mho\mathbf{\Upsilon}^T$ is the eigenvalue decomposition of $\left(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{A}$ and μ_k is the k^{th} eigenvalue. The characteristic polynomial of each block $\mathbf{T}_k$ is

$$
x^2 - (1+\beta-\alpha\mu_k)x + \beta = 0. \tag{3.9}
$$

If $\beta \geq (1-\sqrt{\alpha\mu_k})^2$, then both of the roots in eq. (3.9) will be imaginary and both will have a magnitude $\sqrt{\beta}$. If this condition is satisfied for all $\mathbf{T}_k$, $1 \leq k \leq d$, then the contraction ratio of the whole transformation $\mathbf{T}$ is assured to be $\sqrt{\beta}$. Hence, β can be selected to ensure this upper bound for all eigenvalues. For this purpose, checking only the largest and the smallest μ_k values, which are determined by the MPL in the proof of Theorem 3.1.1 as being $1/(1 \pm \sqrt{\rho})^2$, is sufficient:

$$
\beta \geq \max\left\{ \left| 1 - \frac{\sqrt{\alpha}}{1+\sqrt{r}} \right|, \left| 1 - \frac{\sqrt{\alpha}}{1-\sqrt{r}} \right| \right\}^2 . \tag{3.10}
$$

The lower bound on β can be minimized over α by choosing $\alpha = (1-\rho)^2$, so that the contraction ratio reaches its smallest value of $\sqrt{\beta} = \sqrt{\rho}$. □

If the convergence rates of the damped-IHS and the `M-IHS`, that are given in eq. (3.2) and eq. (3.7), are compared, then an improvement of factor $2/(1+\rho)$ can be observed. In Figure 3.1, the convergence rates of the damped-IHS and the proposed `M-IHS` are numerically compared on a highly over-determined LS problem with size $2^{16} \times 500$. A sketch size $m = 7d$ is used for this experiment. A pseudo-algorithm of the `M-IHS` is given in Algorithm 1.

Iterations of the M-IHS do not require any inner products or norm calculations, which avoids synchronization steps in parallel computing and results in

Algorithm 1 `M-IHS` for the LS problems given in eq. (2.2) with $n \gg d$

```
1: Input: $\mathbf{SA} \in \mathbb{R}^{m \times d}, \mathbf{x}^0, \mathbf{A}, \mathbf{b}$
2: $\beta = d/m$,
3: $\alpha = (1-\beta)^2$
4: while until stopping criteria do
5:     $\mathbf{g}^i = \mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i)$
6:     $(\mathbf{SA})^T(\mathbf{SA})\Delta\mathbf{x}^i = \mathbf{g}^i$        (solve for $\Delta\mathbf{x}^i$)
7:     $\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha\Delta\mathbf{x}^i + \beta(\mathbf{x}^i - \mathbf{x}^{i-1})$
8: end while
```

overwhelming advantages over the CG or the GMRES like iterative solvers in distributed or hierarchical memory systems (see Section 2.4 of [13]). Moreover the `M-IHS` does not need to compute any decomposition or a matrix inversion unlike the state of the art randomized preconditioning techniques such as the Blendenpik and the LSRN. The *Hessian Sketch* (HS) step in *Line 6* of algorithm 1 can be computed inexactly by using a symmetric CG method for a pre-determined tolerance. The inexact scheme is detailed more in Section 3.2.2.

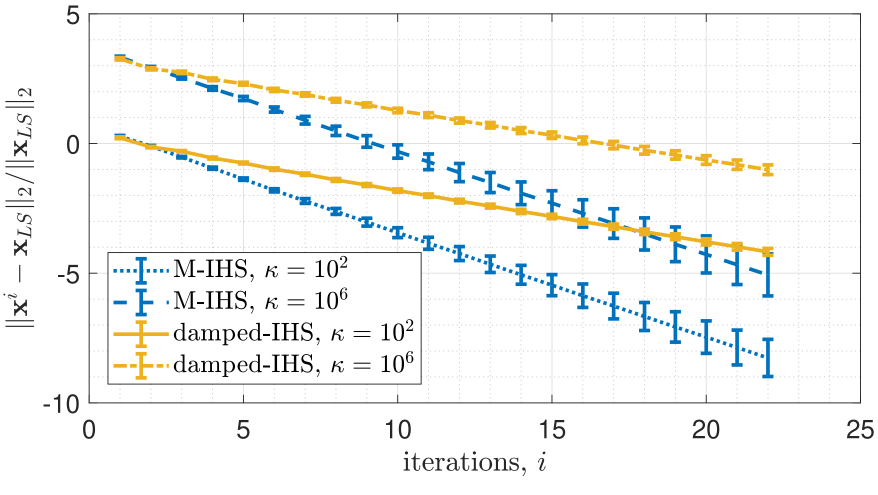


Figure 3.1: *Comparison of the convergence rates: the damped-IHS vs the* `M-IHS`*.*

3.2 Regularized LS Problems: Extensions and Non-asymptotic Analysis of the M-IHS Technique

In this section, we focus on the regularized LS problems given in eq. (2.3). We derive the `Dual M-IHS` technique that is efficient for highly under-determined problems. Then, we establish non-asymptotic convergence analyses of the `M-IHS` and the `Dual M-IHS` techniques. The `Primal Dual M-IHS` is also derived in this section.

3.2.1 M-IHS for regularized LS problems

The `M-IHS` update obtained in eq. (3.5) can be written as two step-update as following:

$$\begin{aligned}
\Delta\mathbf{x}^i &= \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \quad \|\mathbf{S}\mathbf{A}\mathbf{x}\|_2^2 + 2\left\langle \nabla f(\mathbf{x}^i),\ \mathbf{x}\right\rangle, \\
\mathbf{x}^{i+1} &= \mathbf{x}^i + \alpha\Delta\mathbf{x}^i + \beta\left(\mathbf{x}^i - \mathbf{x}^{i-1}\right),
\end{aligned}$$

where $\nabla f(\mathbf{x}^i) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^i - \mathbf{b})$. For regularized LS problems it is modified as:

$$\begin{aligned}
\Delta\mathbf{x}^i &= \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \quad \|\mathbf{S}\mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_2^2 + 2\left\langle \nabla f(\mathbf{x}^i, \lambda),\ \mathbf{x}\right\rangle, \\
\mathbf{x}^{i+1} &= \mathbf{x}^i + \alpha\Delta\mathbf{x}^i + \beta\left(\mathbf{x}^i - \mathbf{x}^{i-1}\right),
\end{aligned} \tag{3.11}$$

where the same sketching matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ is used for all iterations with properly chosen momentum parameters α and β. Here, the linear system is assumed to be strongly over-determined, i.e., $n \gg d$. By using the dual formulation, the theory can be straightforwardly extended to the strongly under-determined case of $d \gg n$ as well [66]. A dual of the problem in eq. (2.3) is

$$\boldsymbol{\nu}(\lambda) = \underset{\boldsymbol{\nu} \in \mathbb{R}^n}{\operatorname{argmin}} \ \underbrace{\frac{1}{2} \left\| \mathbf{A}^T \boldsymbol{\nu} \right\|_2^2 + \frac{\lambda}{2} \left\| \boldsymbol{\nu} \right\|_2^2 - \langle \mathbf{b},\ \boldsymbol{\nu} \rangle,}_{g(\boldsymbol{\nu}, \lambda)} \tag{3.12}$$

and the relation between the solutions of the primal and dual problem is

$$\boldsymbol{\nu}(\lambda) = (\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda))/\lambda \Longleftrightarrow \mathbf{x}(\lambda) = \mathbf{A}^T \boldsymbol{\nu}(\lambda). \tag{3.13}$$

The corresponding `M-IHS` update for the dual problem is:

$$\begin{aligned} \Delta\boldsymbol{\nu}^i &= \underset{\boldsymbol{\nu} \in \mathbb{R}^n}{\operatorname{argmin}} \ \left\| \mathbf{S}\mathbf{A}^T \boldsymbol{\nu} \right\|_2^2 + \lambda \left\| \boldsymbol{\nu} \right\|_2^2 + 2 \left\langle \nabla g(\boldsymbol{\nu}^i, \lambda),\ \boldsymbol{\nu} \right\rangle, \\ \boldsymbol{\nu}^{i+1} &= \boldsymbol{\nu}^i + \alpha \Delta\boldsymbol{\nu}^i + \beta \left(\boldsymbol{\nu}^i - \boldsymbol{\nu}^{i-1} \right), \end{aligned} \tag{3.14}$$

The above update is referred to as `Dual M-IHS`. The primal and dual solutions can be obtained from each other through the relation in eq. (3.13). The convergence rate of the `M-IHS` and the `Dual M-IHS` solvers together with the optimal fixed momentum parameters that maximize the convergence rate are stated in the Theorem 3.2.1 below.

Theorem 3.2.1. *Let* $\mathbf{A}$ *and* $\mathbf{b}$ *be the given data in eq.* (2.1) *with singular values* σ_i *in descending order* $1 \leq i \leq \min(n, d)$, $\mathbf{x}(\lambda) \in \mathbb{R}^d$ *and* $\boldsymbol{\nu}(\lambda) \in \mathbb{R}^n$ *are as in eq.* (2.3) *and eq.* (3.12), *respectively. Let* $\mathbf{U}_1 \in \mathbb{R}^{\max(n,d) \times \min(n,d)}$ *consists of the first* n *rows of an orthogonal basis for* $[\mathbf{A}^T \ \sqrt{\lambda}\mathbf{I}_d]^T$ *if the problem is over-determined, and consists of the first* d *rows of an orthogonal basis for* $[\mathbf{A} \ \sqrt{\lambda}\mathbf{I}_n]^T$ *if the problem is under-determined. Let the sketching matrix* $\mathbf{S} \in \mathbb{R}^{m \times \max(n,d)}$ *be drawn from a distribution* $\mathcal{D}$ *such that*

$$\mathbb{P}_{\mathbf{S} \sim \mathcal{D}} \left(\left\| \mathbf{U}_1^T \mathbf{S}^T \mathbf{S} \mathbf{U}_1 - \mathbf{U}_1^T \mathbf{U}_1 \right\|_2 \geq \epsilon \right) < \delta, \quad \epsilon \in (0, 1). \tag{3.15}$$

Then, the `M-IHS` applied on eq. (2.3) *and the `Dual M-IHS` applied on eq.* (3.12) *with the following momentum parameters*

$$\beta^* = \left(\frac{\epsilon}{1 + \sqrt{1 - \epsilon^2}} \right)^2, \qquad \alpha^* = (1 - \beta^*) \sqrt{1 - \epsilon^2},$$

converge to the optimal solutions, $\mathbf{x}(\lambda)$ *and* $\boldsymbol{\nu}(\lambda)$*, respectively, at the following rate with a probability of at least* $(1-\delta)$*:*

$$\begin{aligned}\left\|\mathbf{x}^{i+1}-\mathbf{x}(\lambda)\right\|_{\mathbf{D}_\lambda^{-1}} &\leq \frac{\epsilon}{1+\sqrt{1-\epsilon^2}}\left\|\mathbf{x}^{i}-\mathbf{x}(\lambda)\right\|_{\mathbf{D}_\lambda^{-1}},\\ \left\|\boldsymbol{\nu}^{i+1}-\boldsymbol{\nu}(\lambda)\right\|_{\mathbf{D}_\lambda^{-1}} &\leq \frac{\epsilon}{1+\sqrt{1-\epsilon^2}}\left\|\boldsymbol{\nu}^{i}-\boldsymbol{\nu}(\lambda)\right\|_{\mathbf{D}_\lambda^{-1}},\end{aligned}$$

where $\mathbf{D}_\lambda^{-1}$ *is the diagonal matrix whose diagonal entries are* $\sqrt{\sigma_i^2+\lambda}$*,* $1 \leq i \leq \min(n,d)$*.*

Proof. In the following proof we denote $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ as the compact SVD with $r = \min(n,d)$. To prove the theorem for the `M-IHS` and the `Dual M-IHS`, we mainly combine the idea of *partly exact* sketching, that is proposed in [36], with the Lyapunov analysis, that we use in the asymptotic analysis of the `M-IHS` in Section 3.1. In parallel to [36], we define the diagonal matrix $\mathbf{D}_\lambda := (\boldsymbol{\Sigma}^2+\lambda\mathbf{I}_r)^{-1/2}$ and the *partly exact* sketching matrix $\mathbf{S}$ as:

$$\widehat{\mathbf{S}} = \begin{bmatrix}\mathbf{S} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_r\end{bmatrix}, \quad \mathbf{S} \in \mathbb{R}^{m\times\max(n,d)}.$$

The proof for M-IHS: Let

$$\widehat{\mathbf{A}} = \begin{bmatrix}\mathbf{U}\boldsymbol{\Sigma}\mathbf{D}_\lambda\\ \sqrt{\lambda}\mathbf{V}\mathbf{D}_\lambda\end{bmatrix} = \begin{bmatrix}\mathbf{U}_1\\ \mathbf{U}_2\end{bmatrix}, \quad \widehat{\mathbf{A}}^T\widehat{\mathbf{A}} = \mathbf{I}_d, \quad \widehat{\mathbf{b}} = \begin{bmatrix}\mathbf{b}\\ \mathbf{0}\end{bmatrix},$$

so that $\mathbf{U}_1$ is the first n rows of an orthogonal basis for $[\mathbf{A}^T \ \ \sqrt{\lambda}\mathbf{I}_d]^T$ as required by the condition in eq. (3.15) of the theorem. To simplify the Lyapunov analysis, the following LS problem will be used:

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathbb{R}^d}{\operatorname{argmin}} \ \left\|\widehat{\mathbf{A}}\mathbf{y}-\widehat{\mathbf{b}}\right\|_2^2 \tag{3.16}$$

which is equivalent to the problem in eq. (2.3) due to the one-to-one mapping $\{\forall\mathbf{x}(\lambda)\in\mathbb{R}^d \mid \mathbf{y}^* = \mathbf{D}_\lambda^{-1}\mathbf{V}^T\mathbf{x}(\lambda)\}$. For the problem in eq. (3.16), the equivalent of the M-IHS given in eq. (3.11) is the following update:

$$\begin{aligned}
\Delta\mathbf{y}^i &= \underset{\mathbf{y}}{\operatorname{argmin}} \ \left\|\widehat{\mathbf{S}}\widehat{\mathbf{A}}\mathbf{y}\right\|_2^2 - 2\langle\widehat{\mathbf{A}}^T(\widehat{\mathbf{b}} - \widehat{\mathbf{A}}\mathbf{y}^i),\ \mathbf{y}\rangle \\
\mathbf{y}^{i+1} &= \mathbf{y}^i + \alpha\Delta\mathbf{y}^i + \beta(\mathbf{y}^i - \mathbf{y}^{i-1})
\end{aligned}$$

with sketched matrix

$$\widehat{\mathbf{S}}\widehat{\mathbf{A}} = \begin{bmatrix}\mathbf{S}\mathbf{U}\mathbf{\Sigma}\mathbf{D}_\lambda \\ \sqrt{\lambda}\mathbf{V}\mathbf{D}_\lambda\end{bmatrix} = \begin{bmatrix}\mathbf{S}\mathbf{U}_1 \\ \mathbf{U}_2\end{bmatrix}.$$

Thus, we can examine the following bipartite transformation to find out the convergence properties of the M-IHS:

$$\begin{bmatrix}\mathbf{y}^{i+1} - \mathbf{y}^* \\ \mathbf{y}^i - \mathbf{y}^*\end{bmatrix} = \underbrace{\begin{bmatrix}(1+\beta)\mathbf{I}_d - \alpha(\widehat{\mathbf{A}}^T\widehat{\mathbf{S}}^T\widehat{\mathbf{S}}\widehat{\mathbf{A}})^{-1} & -\beta\mathbf{I}_d \\ \mathbf{I}_d & \mathbf{0}\end{bmatrix}}_{\mathbf{T}} \begin{bmatrix}\mathbf{y}^i - \mathbf{y}^* \\ \mathbf{y}^{i-1} - \mathbf{y}^*\end{bmatrix}.$$

The contraction ratio of the transformation, which is determined by the eigenvalues, can be found analytically by converting the matrix $\mathbf{T}$ into the following block diagonal form through the same similarity transformation given in eq. (3.8):

$$\mathbf{T} = \mathbf{P}^{-1}\,\mathbf{diag}(\mathbf{T}_1, \ldots, \mathbf{T}_d)\mathbf{P}, \qquad \mathbf{T}_k := \begin{bmatrix}1+\beta-\alpha\mu_k & \beta \\ 1 & 0\end{bmatrix} \tag{3.17}$$

where μ_k is the k^{th} eigenvalue $(\widehat{\mathbf{A}}^T\widehat{\mathbf{S}}^T\widehat{\mathbf{S}}\widehat{\mathbf{A}})^{-1}$. The characteristic polynomials of each block $\mathbf{T}_k$ is

$$x^2 - (1+\beta-\alpha\mu_k)x + \beta = 0, \quad \forall k \in [r].$$

If the following condition holds

$$\beta \geq (1 - \sqrt{\alpha\mu_k})^2, \quad \forall k \in [r], \tag{3.18}$$

then both of the roots are imaginary and both have a magnitude $\sqrt{\beta}$ for all μ_k's. In this case, all linear dynamical systems driven by the above characteristic polynomial will be in the under-damped regime and the contraction rate of the transformation $\mathbf{T}$, through all directions, not just one of them, will be exactly $\sqrt{\beta}$. If the condition in eq. (3.18) is not satisfied for a μ_k with $k \in [r]$, then the linear dynamical system corresponding to that particular λ_k will be in the over-damped regime and the contraction rate in the direction through the eigenvector corresponding to this over-damped system will be smaller compared to the others. As a result, the overall algorithm will be slowed down (see [67] for details). If the condition in eq. (3.15) of Theorem 3.2.1 holds,

$$\left\|\widehat{\mathbf{A}}^T\widehat{\mathbf{S}}^T\widehat{\mathbf{S}}\widehat{\mathbf{A}} - \mathbf{I}_r\right\|_2 = \left\|\mathbf{U}_1^T\mathbf{S}^T\mathbf{S}\mathbf{U}_1 + \mathbf{U}_2^T\mathbf{U}_2 - \mathbf{I}_r\right\|_2 = \left\|\mathbf{U}_1^T\mathbf{S}^T\mathbf{S}\mathbf{U}_1 - \mathbf{U}_1^T\mathbf{U}_1\right\|_2 \leq \epsilon,$$

then, we have the following bounds:

$$\sup_{\|\mathbf{v}\|_2=1} \mathbf{v}^T\widehat{\mathbf{A}}^T\widehat{\mathbf{S}}^T\widehat{\mathbf{S}}\widehat{\mathbf{A}}\mathbf{v} \leq 1+\epsilon \quad \text{and} \quad \inf_{\|\mathbf{v}\|_2=1} \mathbf{v}^T\widehat{\mathbf{A}}^T\widehat{\mathbf{S}}^T\widehat{\mathbf{S}}\widehat{\mathbf{A}}\mathbf{v} \geq 1-\epsilon,$$

which are equivalent to:

$$\underset{k\in[r]}{\text{maximize}}\ \mu_k \leq \frac{1}{1-\epsilon} \quad \text{and} \quad \underset{k\in[r]}{\text{minimize}}\ \mu_k \geq \frac{1}{1+\epsilon}.$$

Consequently, the condition in eq. (3.18) can be satisfied for all μ_k's by the following choice of β that maximizes the convergence rate over step size α

$$\sqrt{\beta^*} = \underset{\alpha}{\text{minimize}} \left(\max\left\{1 - \frac{\sqrt{\alpha}}{\sqrt{1+\epsilon}},\ \frac{\sqrt{\alpha}}{\sqrt{1-\epsilon}} - 1\right\}\right) = \frac{\epsilon}{1+\sqrt{1-\epsilon^2}},$$

where the minimum is achieved at $\alpha^* = \frac{4(1-\epsilon^2)}{(\sqrt{1+\epsilon}+\sqrt{1-\epsilon})^2} = (1-\beta^*)\sqrt{1-\epsilon^2}$ as claimed.

The proof for the Dual M-IHS: The proof of the under-determined case is parallel to the over-determined case except for the following modifications. Let

$$\widehat{\mathbf{A}}^T = \begin{bmatrix} \mathbf{V}\mathbf{\Sigma}\mathbf{D}_\lambda \\ \sqrt{\lambda}\mathbf{U}\mathbf{D}_\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix},\ \widehat{\mathbf{A}}\widehat{\mathbf{A}}^T = \mathbf{I}_n \ \text{ and }\ \widehat{\mathbf{S}}\widehat{\mathbf{A}}^T = \begin{bmatrix} \mathbf{S}\mathbf{V}\mathbf{\Sigma}\mathbf{D}_\lambda \\ \sqrt{\lambda}\mathbf{U}\mathbf{D}_\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{S}\mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix}, \tag{3.19}$$

so that $\mathbf{U}_1$ is the first d rows of an orthogonal basis for $[\mathbf{A} \ \ \sqrt{\lambda}\mathbf{I}_n]$ as required by the theorem. Similar to the `M-IHS` case, the Lyapunov analysis can be simplified by using the following formulation

$$\mathbf{w}^* = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \;\; = \frac{1}{2}\left\|\widehat{\mathbf{A}}^T\mathbf{w}\right\|_2^2 - \langle \mathbf{D}_\lambda \mathbf{U}^T\mathbf{b},\ \mathbf{w}\rangle,$$

which is equivalent to the dual problem in eq. (3.12) due to the one-to-one mapping $\left\{\forall \boldsymbol{\nu}(\lambda) \in \mathbb{R}^n \mid \mathbf{w}^* = \mathbf{D}_\lambda^{-1}\mathbf{U}^T\boldsymbol{\nu}(\lambda)\right\}$. For this form, the equivalent of the `Dual M-IHS` given in eq. (3.14) is

$$\begin{aligned}\Delta\mathbf{w}^i &= \underset{\mathbf{w}}{\operatorname{argmin}} \;\; \left\|\widehat{\mathbf{S}}\widehat{\mathbf{A}}^T\mathbf{w}\right\|_2^2 - 2\langle \mathbf{D}_\lambda\mathbf{U}^T\mathbf{b} - \widehat{\mathbf{A}}\widehat{\mathbf{A}}^T\mathbf{w}^i,\ \mathbf{w}\rangle,\\ \mathbf{w}^{i+1} &= \mathbf{w}^i + \alpha\Delta\mathbf{w}^i + \beta(\mathbf{w}^i - \mathbf{w}^{i-1}).\end{aligned}$$

Therefore, we can analyze the following bipartite transformation to figure out the convergence properties of the `Dual M-IHS`:

$$\begin{bmatrix}\mathbf{w}^{i+1} - \mathbf{w}^*\\ \mathbf{w}^i - \mathbf{w}^*\end{bmatrix} = \underbrace{\begin{bmatrix}(1+\beta)I_n - \alpha(\widehat{\mathbf{A}}\widehat{\mathbf{S}}^T\widehat{\mathbf{S}}\widehat{\mathbf{A}}^T)^{-1} & -\beta\mathbf{I}_n\\ \mathbf{I}_n & \mathbf{0}\end{bmatrix}}_{\mathbf{T}} \begin{bmatrix}\mathbf{w}^i - \mathbf{w}^*\\ \mathbf{w}^{i-1} - \mathbf{w}^*\end{bmatrix}.$$

The rest of the proof can be completed straightforwardly by following the same analysis steps as in the proof for the `M-IHS` case. □

Remark 1. *Theorem 3.2.1 is also valid for the un-regularized problems if, instead of eq.* (3.15)*, the following condition is satisfied*

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}}\left(\left\|\mathbf{U}^T\mathbf{S}^T\mathbf{S}\mathbf{U} - \mathbf{I}_d\right\|_2 \geq \epsilon\right) < \delta, \quad \epsilon \in (0,1), \tag{3.20}$$

The condition in eq. (3.20) *means that the largest and the smallest eigenvalues of* $\left(\mathbf{U}^T\mathbf{S}^T\mathbf{S}\mathbf{U}\right)^{-1}$*, which was asymptotically obtained by the MPL in Section 3.1, should be in the interval* $[(1+\epsilon)^{-1}\ (1-\epsilon)^{-1}]$ *with certain probability. Substituting the bounds of this interval for the eigenvalue estimates obtained by the MPL in Section 3.1 gives the desired result.*

When the necessary conditions are met, the number of iterations needed for both algorithms to reach a certain level of accuracy is stated in the following corollary.

Corollary 3.2.1.1. *For some $\epsilon \in (0,1)/2$ and arbitrary η, if the sketching matrix meets the condition in eq.* (3.15) *and the fixed momentum parameters are chosen as in Theorem 3.2.1, then the number of iterations for the* `M-IHS` *and the* `Dual M-IHS` *to obtain an η-optimal solution approximation in $\ell 2$-norm is upper bounded by*

$$N = \left\lceil \frac{\log(\eta)\log(C)}{\log(\epsilon) - \log(1+\sqrt{1-\epsilon^2})} \right\rceil$$

where the constant C, that is defined as $C = \sqrt{\kappa(\mathbf{A}^T\mathbf{A} + \lambda \mathbf{I}_d)}$ for the `M-IHS` *and $C = \kappa(\mathbf{A})\sqrt{\kappa(\mathbf{A}\mathbf{A}^T + \lambda \mathbf{I}_n)}$ for the* `Dual M-IHS`*, can be removed if the semi-norm in theorem 3.2.1 is used as the solution approximation metric instead of the $\ell 2$ norm.*

Corollary 3.2.1.1 is an immediate result of Theorem 3.2.1. To satisfy the condition in eq. (3.15), a set of cases for the sketching matrix $\mathbf{S}$ are given in Lemma 3.2.2.

Lemma 3.2.2. *If the sketching matrix $\mathbf{S}$ is chosen in one of the following cases, the condition in eq.* (3.15) *of Theorem 3.2.1 is satisfied.*

(i) $\mathbf{S}$ *is a Sparse Subspace Embedding [16] with single nonzero element in each column, with a sketch size*

$$m = \Omega\left(\mathbf{sd}_\lambda(\mathbf{A})^2/(\epsilon^2\delta)\right)$$

where $\Omega(\cdot)$ notation is defined as $a(n) = \Omega(b(n))$, if there exists two integers k and n_0 such that $\forall n > n_0$, $a(n) \geq k \cdot b(n)$. For this case, $\mathbf{SA}$ is computable in $O(\mathbf{nnz}(A))$ operations.

(ii) $\mathbf{S}$ *is a Sparse Subspace Embedding with*

$$s = \Omega(\log_\alpha(\mathsf{sd}_\lambda(\mathbf{A})/\delta)/\epsilon)$$

non-zero elements in each column where $\alpha > 2$, $\delta < 1/2$, $\epsilon < 1/2$, [53, 68], with a sketch size

$$m = \Omega(\alpha \cdot \mathsf{sd}_\lambda(\mathbf{A}) \log(\mathsf{sd}_\lambda(\mathbf{A})/\delta)/\epsilon^2).$$

For this case, $\mathbf{SA}$ *is computable in* $O(s \cdot \mathbf{nnz}(\mathbf{A}))$ *operations.*

(iii) $\mathbf{S}$ *is a SRHT sketching matrix [55, 69] with a sketch size*

$$m = \Omega\left((\mathbf{sd}_\lambda(\mathbf{A}) + \log(1/\epsilon\delta) \log(\mathbf{sd}_\lambda(\mathbf{A})/\delta)) / \epsilon^2 \right).$$

For this case, $\mathbf{SA}$ *is computable in* $O(nd \log(m))$ *operations.*

(iv) $\mathbf{S}$ *is a Sub-Gaussian sketching matrix [14, 69] with a sketch size*

$$m = \Omega(\mathsf{sd}_\lambda(\mathbf{A})/\epsilon^2).$$

For this case, $\mathbf{SA}$ *is computable in* $O(ndm)$ *operations.*

Proof. The following identities will be used on $\mathbf{U}_1$ where $\mathbf{U}, \mathbf{\Sigma}$, and $\mathbf{D}_\lambda$ are defined in the proof of Theorem 3.2.1:

$$\|\mathbf{U}_1\|_F^2 = \|\mathbf{U\Sigma D}_\lambda\|_F^2 = \left\|\mathbf{\Sigma}(\mathbf{\Sigma}^2 + \lambda \mathbf{I}_r)^{-1/2}\right\|_F^2 = \sum_{i=1}^{r} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} = \mathsf{sd}_\lambda(\mathbf{A}),$$

and $\|\mathbf{U}_1\|_2^2 = \frac{\sigma_1^2}{\sigma_1^2+\lambda} \approx 1$ for a properly chosen regularization parameter λ. If the sketch matrix $\mathbf{S}$ is drawn from a randomized distribution $\mathcal{D}$ over matrices $\mathbb{R}^{m \times n}$, then by using the Approximate Matrix Property (AMM) which is given below, it will be proven that the condition in eq. (3.15) can be met with a desired level of probability.

As proven in [69], if a distribution $\mathcal{D}$ over $S \in \mathbb{R}^{m \times n}$ has the $(\epsilon, \delta, 2k, \ell)$-OSE moment property for some $\delta < 1/2$ and $\ell \geq 2$, then it has (ϵ, δ, k)-AMM Property for any $\mathbf{A}, \mathbf{B}$, i.e.,

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}} \left(\left\|\mathbf{A}^T\mathbf{S}^T\mathbf{SB} - \mathbf{A}^T\mathbf{B}\right\|_2 > \epsilon \sqrt{\left(\|\mathbf{A}\|_2^2 + \frac{\|\mathbf{A}\|_F^2}{k} \right) \left(\|\mathbf{B}\|_2^2 + \frac{\|\mathbf{B}\|_F^2}{k} \right)} \right) < \delta. \tag{3.21}$$

The definition of the OSE-moment property can be found in [69]. As it will be detailed next, using the AMM property in eq. (3.21), the sketch sizes in the statement of Lemma 3.2.2 can be found relative to the embedding size k to satisfy the condition in eq. (3.15).

For case (i) of Lemma 3.2.2, Count Sketch with a single nonzero element in each column and size $m \geq 2/(\epsilon'^2\delta)$ has $(\epsilon', \delta, 2)$-JL moment property [70]. JL-Moment Property can be found in Definition 6.1 of [53]. By Theorem 6.2 in [53]:

$$\left\|\mathbf{U}_1\mathbf{S}^T\mathbf{S}\mathbf{U}_1 - \mathbf{U}_1^T\mathbf{U}_1\right\|_F < 3\epsilon' \left\|\mathbf{U}_1\right\|_F^2 = 3\epsilon'\mathsf{sd}_\lambda(\mathbf{A}) \leq \epsilon$$

for $\epsilon' = \epsilon/(3\mathsf{sd}_\lambda(\mathbf{A}))$. So, condition in eq. (3.15) holds with probability at least $1 - \delta$, if $m = O(\mathsf{sd}_\lambda(\mathbf{A})^2/(\epsilon^2\delta))$.

For case (ii) of Lemma 3.2.2, combining Theorem 4.2 of [71] and Remark 2 of [69] implies that any sketch matrix drawn from an OSNAP [68] with the conditions given in case (ii) of Lemma 3.2.2 satisfies the $(\epsilon', \delta, k, \log(k/\delta))$-OSE moment property thus the $(\epsilon', \delta, k/2)$-AMM Property. Setting $\mathbf{A} = \mathbf{B} = \mathbf{U}_1$ and $k = \mathsf{sd}_\lambda(\mathbf{A})/2$ in eq. (3.21) gives:

$$\left\|\mathbf{U}_1^T\mathbf{S}^T\mathbf{S}\mathbf{U}_1 - \mathbf{U}_1^T\mathbf{U}_1\right\|_2 \leq \epsilon'(\left\|\mathbf{U}_1\right\|_2^2 + 2) \leq 3\epsilon' \leq \epsilon$$

with probability of at least $(1 - \delta)$.

Remark 2. *Based on the lower bounds established for any OSE in [72], the Conjecture 14 in [68] states that any OSNAP with $m = \Omega((k + \log(1/\delta))/\epsilon^2)$ and $s = \Omega(\log(k/\delta)/\epsilon)$ have the $(\epsilon, \delta, k, \ell)$-OSE moment property for $\ell = \Theta(\log(k/\delta))$, an even integer. If this conjecture is proved, then by the AMM property in eq.* (3.21)*, the condition in eq.* (3.15) *can be satisfied with probability at least $(1 - \delta)$ by using an OSNAP matrix with size $m = \Omega((\mathsf{sd}_\lambda(A) + \log(1/\delta))/\epsilon^2)$ and sparsity $s = \Omega(\log(\mathsf{sd}_\lambda(A)/\delta)/\epsilon)$.*

For case (iii) of lemma 3.2.2, by Theorem 9 of [69], SRHT with the sketch size given in case (iii) has the $(\epsilon', \delta, 2\mathsf{sd}_\lambda(\mathbf{A}), \log(\mathsf{sd}_\lambda(\mathbf{A})/\delta))$-OSE moment property and thus it provides $(\epsilon', \delta, \mathsf{sd}_\lambda(\mathbf{A}))$-AMM property. Again, setting $\mathbf{A} = \mathbf{B} = \mathbf{U}_1$ and $k = \mathsf{sd}_\lambda(\mathbf{A})$ in eq. (3.21) produces the desired result.

For case (iv) of Lemma 3.2.2, the Subgaussian matrices having entries with mean zero and variance $1/m$ satisfy the JL Lemma [73] with optimal sketch size [53]. Also, they have the $(\epsilon/2, \delta, \Theta(\log(1/\delta)))$-JL moment property [74]. Thus by Lemma 4 of [69] such matrices have $(\epsilon, \delta, k, \Theta(k + \log(1/\delta)))$-OSE moment property for $\delta < 9^{-k}$, which means $m = \Omega(k/\epsilon^2)$. Again, by setting $\mathbf{A} = \mathbf{B} = \mathbf{U}_1$ and $k = \mathsf{sd}_\lambda(\mathbf{A})$ in eq. (3.21) produces the desired result. □

Lemma 3.2.2 suggests that in order to satisfy the condition in Theorem 3.2.1, the sketch size can be chosen proportional to the statistical dimension of the coefficient matrix which can be considerably smaller than its rank. Moreover, to obtain a solution approximation, the second condition in Lemma 11 of [36] is not a requirement, hence we obtained slightly better results for the sparse subspace embeddings in the cases of (i) and (ii) of Lemma 3.2.2. In the following Corollary 3.2.2.1, we obtained substantially simplified empirical versions of the convergence rate, momentum parameters and required sketch size by using the MPL and approximating the filtering coefficients of Tikhonov regularization with binary coefficients. Corollary 3.2.2.1 suggests that the ratio between the statistical dimension and the sketch size determines the convergence rate of the proposed algorithms, which interestingly seems valid even for the sketch matrices with a single non-zero element in each column.

Corollary 3.2.2.1. *If the entries of the sketching matrix are independent, zero mean, unit variance with bounded higher order moments, and the Truncated SVD regularization with truncation parameter $\lceil \mathsf{sd}_\lambda(\mathbf{A}) \rceil$ is used, then the `M-IHS` and the `Dual M-IHS` with the following momentum parameters*

$$\beta = \frac{\mathsf{sd}_\lambda(\mathbf{A})}{m}, \qquad \alpha = (1-\beta)^2$$

will converge to the optimal solutions with a convergence rate of $\sqrt{\beta}$ as $m \to \infty$ while $\mathsf{sd}_\lambda(\mathbf{A})/m$ remains constant. Any sketch size $m > \mathsf{sd}_\lambda(\mathbf{A})$ can be chosen to obtain an η-optimal solution approximation in most $\frac{\log(\eta)}{\log(\sqrt{\beta})}$ iterations.

Proof. Consider the regularized LS solution with parameter λ and the Truncated SVD solution with parameter $\lceil \mathsf{sd}_\lambda(\mathbf{A}) \rceil$:

$$\mathbf{x}(\lambda) = \sum_{i=1}^{r} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i \quad \text{and} \quad \mathbf{x}^\dagger = \sum_{i=1}^{\lceil \mathsf{sd}_\lambda(\mathbf{A}) \rceil} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i \tag{3.22}$$

where $\mathbf{u}_i$'s and $\mathbf{v}_i$'s are columns of $\mathbf{U}$ and $\mathbf{V}$ matrices in the SVD. The Tikhonov regularization with the closed form solution is preferred in practice to avoid the high computational cost of the SVD. The filtering coefficients of the Tikhonov regularization, $\frac{\sigma_i^2}{\sigma_i^2+\lambda}$, become very close to the binary filtering coefficients of the TSVD, as the decay rate of the singular values of $\mathbf{A}$ increases. In these cases, $\mathbf{x}(\lambda)$ and $\mathbf{x}^\dagger$ in eq. (3.22) are very close to each other (Section 4 and 5 of [38]). Thereby, the diagonal matrix $\mathbf{\Sigma D}$ which is used in the proof of Lemma 3.2.2 can be approximated by the diagonal matrix $\mathbf{\Gamma}$ where

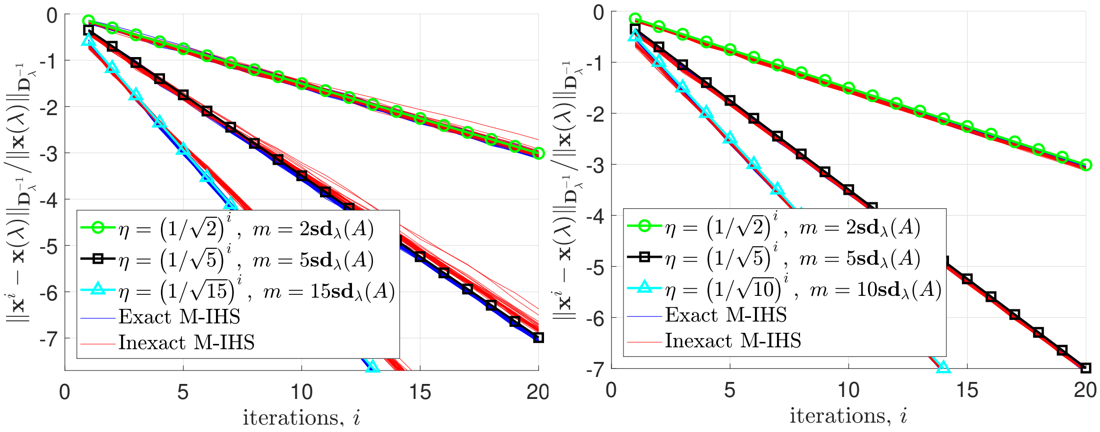
$$\mathbf{\Gamma}_{ii} = \begin{cases} 1 & \text{if } i \leq \mathsf{sd}_\lambda(\mathbf{A}) \leq r \\ 0 & \textit{otherwise} \end{cases} \quad ,$$

which is equivalent to replacing the Tikhonov coefficients by the binary coefficients. Then, we have the following close approximation:

$$\begin{aligned} \left(\widehat{\mathbf{A}}^T \widehat{\mathbf{S}}^T \widehat{\mathbf{S}} \widehat{\mathbf{A}}\right)^{-1} &= \left(\mathbf{D\Sigma U}^T \mathbf{S}^T \mathbf{S U \Sigma D} + \lambda \mathbf{D}^2\right)^{-1} \\ &\approx \left(\mathbf{\Gamma}(\mathbf{SU})^T(\mathbf{SU})\mathbf{\Gamma} + \mathbf{I}_r - \mathbf{\Gamma}\right)^{-1} = \left[\begin{array}{c|c} \overline{\mathbf{S}}^T \overline{\mathbf{S}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_{(r-\mathsf{sd}_\lambda(\mathbf{A}))} \end{array}\right]^{-1}, \end{aligned}$$

where $\overline{\mathbf{S}} = \mathbf{SU\Gamma} \in \mathbb{R}^{m \times \mathsf{sd}_\lambda(\mathbf{A})}$ has the same distribution as $\mathbf{S}$, since $\mathbf{U\Gamma}$ is an orthonormal transformation. By the MPL, the minimum and the maximum eigenvalues of this approximation is asymptotically bounded in the interval $[\left(1+\sqrt{\mathsf{sd}_\lambda(\mathbf{A})/m}\right)^{-2}, \left(1-\sqrt{\mathsf{sd}_\lambda(\mathbf{A})/m}\right)^{-2}]$ as $m \to \infty$ and while $\mathsf{sd}_\lambda(\mathbf{A})/m$ remains constant [63]. The rest of the proof follows from the analysis given in the proof of Theorem 3.2.1. □

Although the MPL provides bounds for the singular values of the sketching matrix $\mathbf{S}$ in the asymptotic regime, i.e., as $m \to \infty$; these bounds become very



(a) *Dense problem, SRHT sketch via DCT* (b) *Sparse problem, Count sketch*

Figure 3.2: *Comparison of the theoretical rate given in Corollary 3.2.2.1 and the empirical convergence rate. The lines with different markers show the theoretical convergence rate for different sketch sizes. Both the exact and the inexact (given in Algorithm 2) versions of the* `M-IHS` *were run* 32 *times and the result of each run is plotted as a separate line. Except for a small degradation in the dense case, setting the forcing term to a small constant such as* $\epsilon_{sub} = 0.1$ *is sufficient for the inexact scheme to achieve the same rate as the exact version in these experiments.*

good estimators of the actual bounds when m takes finite values, as demonstrated in Figure 3.2. In Figure 3.2a, $\mathbf{A} \in \mathbb{R}^{32768\times1000}$ with $\kappa(\mathbf{A}) = 10^8$ was generated as described in Section 3.4.1. In Figure 3.2b, $\mathbf{A} \in \mathbb{R}^{24336\times1296}$ was generated by using *sprand* command of MATLAB. We first created a sparse matrix with size $\tilde{\mathbf{A}} \in \mathbb{R}^{20\times6}$ and sparsity of 15%, then the final form was obtain by taking $\mathbf{A} = \tilde{\mathbf{A}}^{\otimes4}$ and deleting the all-zero rows. The final form of $\mathbf{A}$ has a sparsity ratio of 0.1% and the condition number of $\kappa(\mathbf{A}) = 10^7$. The noise level was set to 1% and the regularization parameter λ that minimizes the error $\|\mathbf{x}_0 - \mathbf{x}(\lambda)\|$ was used in both experiments. The resulting statistical dimensions were 119 and 410, respectively. The rate of $\sqrt{\beta}$ in Corollary 3.2.2.1 creates a remarkable fit to the numerical convergence rate of the `M-IHS` variants when the momentum parameters given in Corollary 3.2.2.1 are used even for the Tikhonov regularization. This is because the sigmoid-like filtering coefficients in the Tikhonov regularization can be thought of as the smoothed version of the binary coefficients in the TSVD solution and therefore the binary coefficients constitute a good approximation for the filtering coefficients of the Tikhonov regularization.

Remark 3. *The momentum parameters given in Corollary 3.2.2.1 maximizes the convergence rate when the statistical dimension is known. If* $\mathsf{sd}_\lambda(\mathbf{A})$ *is overestimated and thus* β *is chosen larger than the ratio* $\mathsf{sd}_\lambda(\mathbf{A})/m$ *and* $\alpha = (1-\beta)^2$*, then the convergence rate is still* $\sqrt{\beta}$ *since the dynamical system will be still in the under-damped regime. An empirical algorithm to estimate* $\mathsf{sd}_\lambda(\mathbf{A})$ *by using the Hutchinson-like estimators is detailed in Section 3.2.4.*

3.2.2 Efficient M-IHS sub-solvers

In practice, the `M-IHS` and the `Dual M-IHS` eliminate the dominant term $O(mr^2)$ in the complexity expression of well known solvers such as the Blendenpik and the LSRN by approximately solving the lower dimensional linear systems in eq. (3.11) and eq. (3.14) avoiding matrix decompositions or inversions. This *inexact sub-solver* approach provides a trade-off opportunity between the computational complexity and the convergence rate, which is highly desirable in very large dimensional problems. Unfortunately, such a trade-off is not possible for the Blendenpik and LSRN techniques which require a full matrix decomposition. Inexact sub-solvers have been known to be a good heuristic way to create this trade-off and they are widely used in the algorithms that are based on the Newton Method to solve the large scale normal equations [75]. In these inexact (or truncated) Newton Methods, inner iterations are terminated at the moment that the relative residual error is lower than an iteration-dependent threshold, named as the *forcing terms* [76]. In the literature, there are various techniques to choose these forcing terms that guarantee a global convergence [77], but the number of iterations suggested by these techniques are significantly higher than the total number of iterations used in practice. Therefore, in this work the heuristic constant threshold ϵ_{sub}, that checks the relative residual error of the linear system [78], is used.

Efficient but approximate solutions to the sub-problems in eq. (3.5) and eq. (3.14) can be obtained by Krylov subspace based first order methods. However, LSQR-like solvers that are adapted for the normal equations would require

Algorithm 2 `M-IHS` (for $n \geq d$)

```
Input: $\mathbf{A}$, $\mathbf{b}$, $m$, $\lambda$, $\mathbf{x}^1$, $\mathtt{sd}_\lambda(\mathbf{A})$, $\epsilon_{sub}$        complexity
$\mathbf{SA} = \mathtt{RP\_fun}(\mathbf{A}, m)$        $C(m,n,d)$
  $\beta = \mathtt{sd}_\lambda(\mathbf{A})/m$        $O(1)$
  $\alpha = (1-\beta)^2$        $O(1)$
while until stopping criteria do
    $\mathbf{g}^i = \mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i) - \lambda\mathbf{x}^i$        $O(nd)$
    $\Delta\mathbf{x}^i = \mathtt{AAb\_Solver}(\mathbf{SA},\ \mathbf{g}^i,\ \lambda,\ \epsilon_{sub})$        $O(md)$
    $\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha\Delta\mathbf{x}^i + \beta(\mathbf{x}^i - \mathbf{x}^{i-1})$        $O(d)$
end while
```

Algorithm 3 `Dual M-IHS` (for $n \leq d$)

```
Input: $\mathbf{A}$, $\mathbf{b}$, $m$, $\lambda$, $\mathtt{sd}_\lambda(\mathbf{A})$, $\epsilon_{sub}$        complexity
$\mathbf{SA}^T = \mathtt{RP\_fun}(\mathbf{A}^T, m)$        $C(m,n,d)$
  $\beta = \mathtt{sd}_\lambda(\mathbf{A})/m$        $O(1)$
  $\alpha = (1-\beta)^2$        $O(1)$
  $\boldsymbol{\nu}^0 = 0$        $O(1)$
while until stopping criteria do
    $\mathbf{g}^i = \mathbf{b} - \mathbf{A}\mathbf{A}^T\boldsymbol{\nu}^i - \lambda\boldsymbol{\nu}^i$        $O(nd)$
    $\Delta\boldsymbol{\nu}^i = \mathtt{AAb\_Solver}(\mathbf{SA}^T,\ \mathbf{g}^i,\ \lambda,\ \epsilon_{sub})$        $O(mn)$
    $\boldsymbol{\nu}^{i+1} = \boldsymbol{\nu}^i + \alpha\Delta\boldsymbol{\nu}^i + \beta(\boldsymbol{\nu}^i - \boldsymbol{\nu}^{i-1})$        $O(n)$
end while
$\mathbf{x}^{N+1} = \mathbf{A}^T\boldsymbol{\nu}^{N+1}$        $O(nd)$
```

computations of 4 matrix-vector multiplications per iteration. On the other hand, due to the explicit calculation of $(\mathbf{SA})^T(\mathbf{SA})\mathbf{z}$, the symmetric CG, that would require only 2 matrix-vector multiplications per iteration, might be unstable for the ill-conditioned problems [49]. Therefore, in Section 3.3, we propose a stable sub-solver, referred to as `AAb_Solver`, which is particularly designed for the problems in the form of $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{b}$. The `AAb_Solver` is based on the GKL Bidiagonalization and it uses a similar approach that the LSQR uses on the LS problems. In addition to the stability advantage over the symmetric CG technique, `AAb_Solver` produces a bidiagonal representation of sketched matrix as a byproduct of the iterations. This bidiagonal form can be exploited to estimate parameters including λ and $\mathtt{sd}_\lambda(\mathbf{A})$ [10]. The inexact versions of the `M-IHS` and the `Dual M-IHS` that use `AAb_Solver` are given in Algorithm 2 and Algorithm 3, where `RP_fun` represents the function that generates the desired sketched matrix such that $\mathbb{E}\left[\mathbf{S}^T\mathbf{S}\right] = \mathbf{I}_m$ whose implementation details can be found in the relevant references in Lemma 3.2.2. Number of operations required at each step is stated at the right most column of the algorithms, where $C(\cdot)$ represents the complexity of constructing the sketching matrix as given in Lemma 3.2.2. Setting the forcing term ϵ_{sub}, for instance, to 0.1 for all iterations is enough for the inexact `M-IHS` variants to converge at the same rate $\sqrt{\beta}$ as the exact versions as demonstrated in Figure 3.2.

3.2.3 Two-stage sketching for the M-IHS variants

Lemma 3.2.2 suggests that if the statistical dimension is several times smaller than the dimensions of $\mathbf{A}$, then it is possible to choose a substantially smaller sketch size than $\min(n, d)$. If this is the case, then the quadratic objective functions in eq. (3.11) and eq. (3.14) become strongly under-determined problems, which makes it possible to approximate the Hessian of the objective functions one more time by taking their convex dual as it has been done in the `Dual M-IHS`. This approach is similar to the approach where the problems in eq. (3.11) and eq. (3.14) are approximately solved by using the `AAb_Solver`, with an additional dimension reduction. As a result of two Hessian sketching, the linear sub-problem whose dimensions are reduced from both sides can be efficiently solved by the `AAb_Solver` for a pre-determined tolerance as before. For the details of this two-step approach, consider the following dual of the sub-problem in eq. (3.5)

$$\mathbf{z}^* = \underset{\mathbf{z}\in\mathbb{R}^m}{\operatorname{argmin}} \quad \underbrace{\frac{1}{2}\left\|\mathbf{A}^T\mathbf{S}^T\mathbf{z} + \nabla f(\mathbf{x}^i, \lambda)\right\|_2^2 + \frac{\lambda}{2}\left\|\mathbf{z}\right\|_2^2}_{h(\mathbf{z},\mathbf{x}^i,\lambda)}, \tag{3.23}$$

which is a strongly over-determined problem if $m \ll \min(n, d)$. Hence, it can be approximately solved by the M-IHS updates as

$$\begin{aligned}\Delta\mathbf{z}^j &= \underset{z\in\mathbb{R}^m}{\operatorname{argmin}} \quad \left\|\mathbf{W}\mathbf{A}^T\mathbf{S}^T\mathbf{z}\right\|_2^2 + \lambda\left\|\mathbf{z}\right\|_2^2 + 2\left\langle\nabla_{\mathbf{z}}h(\mathbf{z}^j, \mathbf{x}^i, \lambda),\ \mathbf{z}\right\rangle, \\ \mathbf{z}^{j+1} &= \mathbf{z}^j + \alpha_2\Delta\mathbf{z}^j + \beta_2\left(\mathbf{z}^j - \mathbf{z}^{j-1}\right).\end{aligned} \tag{3.24}$$

After M iterations, the solution of eq. (3.5) can be recovered by using the relation in eq. (3.13) as $\Delta\mathbf{x}^i = (\nabla f(\mathbf{x}^i, \lambda) - \mathbf{A}^T\mathbf{S}^T\mathbf{z}^M)/\lambda$. The same strategy can be applied on the sub-problem in eq. (3.14) by replacing $\mathbf{SA}$ with $\mathbf{SA}^T$ and $\nabla f(\mathbf{x}^i, \lambda)$ with $\nabla g(\boldsymbol{\nu}^i, \lambda)$. The resulting algorithms, referred to as `Primal Dual M-IHS`, are given in Algorithm 4 and Algorithm 5, respectively.

Algorithm 4 `Primal Dual M-IHS` (for $n \leq d$)

```
Input: $\mathbf{A},\ \mathbf{b},\ m_1,\ m_2,\ \lambda,\ \mathsf{sd}_\lambda(\mathbf{A}),\ \epsilon_{sub}$                 complexity
    $\mathbf{SA}^T = \texttt{RP\_fun}(\mathbf{A}^T, m_1)$                                     $C(m_1, n, d)$
  $\mathbf{WAS}^T = \texttt{RP\_fun}(\mathbf{SA}^T, m_2)$                                     $C(m_1, m2, n)$
      $\beta_\ell = \mathsf{sd}_\lambda(\mathbf{A})/m_\ell, \quad \ell = 1, 2$                              $O(1)$
      $\alpha_\ell = (1-\beta_\ell)^2, \quad \ell = 1, 2$                                  $O(1)$
    $\boldsymbol{\nu}^{1,0} = 0,\ \mathbf{z}^{1,0} = 0$                                        $O(1)$
for i=1:N do
    $\mathbf{b}^i = \mathbf{b} - \mathbf{AA}^T\boldsymbol{\nu}^i - \lambda\boldsymbol{\nu}^i$                                  $O(nd)$
    for j=1:M do
        $\mathbf{g}^{i,j} = \mathbf{SA}^T(\mathbf{b}^i - \mathbf{AS}^T\mathbf{z}^{i,j}) - \lambda\mathbf{z}^j$                     $O(nm_1)$
        $\Delta\mathbf{z}^{i,j} = \texttt{AAb\_Solver}(\mathbf{WAS}^T, \mathbf{g}^{i,j},\ \lambda,\ \epsilon_{sub})$           $O(m_1m_2)$
        $\mathbf{z}^{i,j+1} = \mathbf{z}^{i,j} + \alpha_2\Delta\mathbf{z}^{i,j} + \beta_2(\mathbf{z}^{i,j} - \mathbf{z}^{i,j-1})$           $O(m_1)$
    end for
    $\Delta\boldsymbol{\nu}^i = (\mathbf{b}^i - \mathbf{AS}^T\mathbf{z}^{i,M+1})/\lambda, \quad \mathbf{z}^{i+1,0} = \mathbf{z}^{i,M+1}$          $O(nm_1)$
    $\boldsymbol{\nu}^{i+1} = \boldsymbol{\nu}^i + \alpha_1\Delta\boldsymbol{\nu}^i + \beta_1(\boldsymbol{\nu}^i - \boldsymbol{\nu}^{i-1})$                     $O(n)$
end for
$\mathbf{x}^{N+1} = \mathbf{A}^T\boldsymbol{\nu}^{N+1}$                                             $O(nd)$
```

The two-stage sketching idea presented here is first suggested by Zhang et al. in [1]. They used the A-IHS technique to solve the sub-problems that arise during the iterations of the Accelerated Iterative Dual Random Projection (A-IDRP) which is a dual version of the A-IHS. However, since both of the A-IHS and the

Algorithm 5 `Primal Dual M-IHS` (for $n \geq d$)

```
1:  Input: $\mathbf{A}$, $\mathbf{b}$, $m_1$, $m_2$, $\lambda$, $x^1$, $\mathtt{sd}_\lambda(\mathbf{A})$, $\epsilon_{sub}$          complexity
2:        $\mathbf{SA} = \mathtt{RP\_fun}(\mathbf{A}, m_1)$                                                   $C(m_1, n, d)$
3:  $\mathbf{WA}^T\mathbf{S}^T = \mathtt{RP\_fun}(\mathbf{SA}, m_2)$                                             $C(m_1, m_2, d)$
4:        $\beta_\ell = \mathtt{sd}_\lambda(\mathbf{A})/m_\ell, \quad \ell = 1, 2$                                       $O(1)$
5:        $\alpha_\ell = (1 - \beta_\ell)^2, \quad \ell = 1, 2$                                          $O(1)$
6:        $\mathbf{x}^0 = 0,\ \mathbf{z}^{1,0} = 0$                                                     $O(1)$
7:  for i=1:N do
8:    $\mathbf{b}^i = \mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i) - \lambda\mathbf{x}^i$                                     $O(nd)$
9:    for j=1:M do
10:       $\mathbf{g}^{i,j} = \mathbf{SA}(\mathbf{b}^i - \mathbf{A}^T\mathbf{S}^T\mathbf{z}^{i,j}) - \lambda\mathbf{z}^{i,j}$                              $O(dm_1)$
11:      $\Delta\mathbf{z}^{i,j} = \mathtt{AAb\_Solver}(\mathbf{WA}^T\mathbf{S}^T, \mathbf{g}^{i,j},\ \lambda,\ \epsilon_{sub})$                 $O(m_1m_2)$
12:      $\mathbf{z}^{i,j+1} = \mathbf{z}^{i,j} + \alpha_2\Delta\mathbf{z}^{i,j} + \beta_2(\mathbf{z}^{i,j} - \mathbf{z}^{i,j-1})$                      $O(m_1)$
13:   end for
14:    $\Delta\mathbf{x}^i = (\mathbf{b}^i - \mathbf{A}^T\mathbf{S}^T\mathbf{z}^{i,M+1})/\lambda, \quad \mathbf{z}^{i+1,0} = \mathbf{z}^{i,M+1}$                 $O(dm_1)$
15:   $\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha_1\Delta\mathbf{x}^i + \beta_1(\mathbf{x}^i - \mathbf{x}^{i-1})$                            $O(d)$
16: end for
```

A-IDRP are based on the CG technique, the convergence rate of the proposed A-IHS, A-IDRP and the primal dual algorithm called as Accelerated Iterative Primal Dual Sketch (A-IPDS) are all degraded in the LS problems with high condition numbers due to the instability issue of the symmetric CG technique [49]. Even if the regularization is used, still the performance of the solvers proposed in [1] are considerably deteriorated compared to the other randomized preconditioning techniques as shown in Section 3.4. Further, applying the preconditioning idea of IHS to the stable techniques such as the LSQR that are adapted for the LS problem is not so efficient as the `M-IHS` variants, because they require two preconditioning systems to be solved per iteration.

The computational saving when we apply a second dimension reduction as in the `Primal Dual M-IHS` may not be significant due to the second gradient computations in *Line 10* of the given algorithms, but the lower dimensional subproblems that we obtain at the end of the second sketching can be used to estimate several parameters including the regularization parameter as detailed in Chapter 4.

The `Primal Dual M-IHS` techniques are extension of the inexact schemes. Therefore, their convergence rates depend on their forcing terms that are used to stop the inner iterations [77]. In [1], an upper bound for the error of the primal dual updates is proposed. However as it is detailed in Appendix A, there are several inaccuracies in the development of the bound. Therefore, finding a provably valid lower bound on the number of inner loop iterations, that guarantee a certain rate of convergence at the main loop, is still an open problem for the primal dual algorithms.

3.2.4 Estimation of the statistical dimension

The statistical dimension $\mathsf{sd}_\lambda(\mathbf{A})$ in Algorithm 2, 3, 4 and 5 can be estimated by using a Hutchinson-like randomized trace estimator [79]. Alternatively, $\mathsf{sd}_\lambda(\mathbf{A})$ can be estimated by using the algorithm proposed in [36] within a constant factor in nnz$(\mathbf{A})$ time with a constant probability, if $\mathsf{sd}_\lambda(\mathbf{A}) \leq \xi$ where:

$$\xi = \min\{n, d, \lfloor (n+d)^{1/3}/\text{poly}(\log(n+d))\rfloor\}.$$

However, due to the third order root and the division by typically higher than a sixth order polynomial, ξ becomes very small and the proposed algorithm in [36] can only be used when the singular values of $\mathbf{A}$ decay severely/exponentially. Therefore, we preferred to use the heuristic trace estimator in Algorithm 6, where the input matrix $\mathbf{SA}$ can be replaced with $\mathbf{SA}^T$ or even with $\mathbf{WA}^T\mathbf{S}^T$ and $\mathbf{WAS}^T$ according to the requirements of the algorithm used. Any estimator in [79] can be substituted for the Hutchinson Estimator and the number of samples T can be chosen accordingly. In the conducted experiments with various singular value profiles, small samples sizes such as 2 or 3 and $\epsilon_{tr} = 0.5$ was sufficient to obtain satisfactory estimates for $\mathsf{sd}_\lambda(\mathbf{A})$ used in Corollary 3.2.2.1. Note that, as long as $\mathsf{sd}_\lambda(\mathbf{A})$ is overestimated, the convergence rates of the proposed algorithms will be strictly controlled by β as in Corollary 3.2.2.1.

Algorithm 6 `Inexact Hutchinson Trace Estimator`

```
Input: $\mathbf{SA} \in \mathbb{R}^{m \times d}$, $\lambda$, $T$, $\epsilon_{tr}$        complexity
$\mathbf{v}^\ell = \{-1, +1\}^d, \quad \ell = 1, \ldots, T$        $O(Td)$
$\tau = 0$        $O(1)$
for i = 1:T do
    $\mathbf{z}^i = \texttt{AAb\_Solver}(\mathbf{SA}, \mathbf{v}^i, \lambda, \epsilon_{tr})$        $O(md)$
    $\tau = \tau + \lambda \langle \mathbf{v}^i, \ \mathbf{z}^i \rangle$        $O(d)$
end for
Output: $\widehat{\mathsf{sd}_\lambda} = d - \tau / T$        $O(1)$
```

3.2.5 Complexity analyses of the proposed algorithms

The iterations of both the exact and inexact `M-IHS` and `Dual M-IHS` consist of 4 stages with the following computational complexities:

Stage	**Exact schemes**	**Inexact schemes**
generation of SA or SA^T	$C(n, d, m)$	$C(n, d, m)$
QR (**R** – **factor only)**	$O(mr^2)$	–
$\mathsf{sd}_\lambda(\mathbf{A})$ **est.**	$O(Tr^2)$	$O_{\epsilon_{tr}}(Tmr)$
N **iterations**	$O(N(nd + r^2))$	$O(Nnd) + O_{\epsilon_{sub}}(Nmr)$

Here, N is the number of iterations, T is the number of samples used in Hutchinson-like estimators, $r = \min(n, d)$ and $C(\cdot)$ is the complexity of generating the sketched matrix which is noted in Lemma 3.2.2. Also, we assumed that the sub-problems in eq. (3.5) and eq. (3.14) are solved by using the QR decomposition for the exact schemes. Notation $O_\epsilon(\cdot)$ is used to indicate that complexity depends on the tolerance ϵ that is used to terminate the sub-solver iterations. The major advantage of the proposed techniques over the current RP solvers is the ability of avoiding the complexity of the QR step. In the inexact `M-IHS` variants, the third order complexity $O(mr^2)$ of matrix decomposition or inversion are avoided. For the applications where m grows larger, this saving become critical as shown in Section 3.4.5. Since the sketch size m can be chosen proportional to the statistical dimension, the memory space required by all the above techniques is $O(\mathsf{sd}_\lambda(\mathbf{A})r)$.

In a similar manner, the complexity of each stage in the `Primal Dual M-IHS` variants is:

Stage	**Exact schemes**	**Inexact schemes**
sketching	$C(n,d,m_1)+C(r,m_1,m_2)$	$C(n,d,m_1)+C(r,m_1,m_2)$
QR or SVD	$O(m_2m_1^2)$	$-$
$\mathsf{sd}_\lambda(\mathbf{A})$ **est.**	$O(Tm_1^2)$	$O_{\epsilon_{tr}}(Tm_1m_2)$
N **iterations**	$O(Nnd+NM(m_1r+m_1^2))$	$O(N(nd+Mm_1r))+O_{\epsilon_{sub}}(NMm_1m_2)$

Here, N and M denote number of outer and inner iterations, respectively. Unless $Mm_1 \ll r$, the `Primal Dual M-IHS` variants do not provide significant saving over the `M-IHS` or the `Dual M-IHS`. However, when n and d scale similar and the ratio $\mathsf{sd}_\lambda(\mathbf{A})/r$ is very small, if the decomposition of the sketched matrix is required for parameter estimation purpose as discussed in Chapter 4, then due to the decomposition of $m_2 \times m_1$-dimensional doubly sketched matrix, the `Primal Dual M-IHS` variants require far fewer operations then any exact schemes which need to compute the decomposition of $r \times m_1$ dimensional sketched matrix. Such conditions are prevalent, for example, in image de-blurring or seismic travel-time tomography problems [11]. The memory space required by `Primal Dual M-IHS` techniques is $O(m_1r+m_1m_2)$.

Depending on the type of choice of the sketching used, the complexity of the proposed techniques vary significantly. For dense coefficient matrices while the SRHT matrices has lower run time in sequential environments, Gaussian matrices would be more efficient in parallel computing. If the coefficient matrix is sparse, then the data oblivious sketching types such as OSNAP or CountSketch matrices would be effective choices with run time of $O(\text{nnz}(\mathbf{A}))$. The proposed techniques can be still used even if the coefficient matrix is an operator, in this case Gaussian or sparse embeddings can be utilized. If the coefficient matrix is sparse or an operator that allows fast matrix-vector computations, then both exact and inexact schemes are automatically sped up due to the saving in the gradient computation. For instance, in the sparse case, complexity of the gradient computation is reduced from $O(nd)$ to $O(\text{nnz}(\mathbf{A}))$.

3.3 A solver for linear systems in the form of $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{b}$

The linear sub-problems in the form of $(\mathbf{A}^T\mathbf{A}+\lambda\mathbf{I})\mathbf{x} = \mathbf{b}$, whose solutions are required by all four of the proposed `M-IHS` variants, can be approximately solved by using the *bidiag2* procedure described in [49], which produces an upper bidiagonal matrix as:

$$\mathbf{P}_k^T\mathbf{A}\mathbf{Q}_k = \mathbf{R}_k = \begin{bmatrix} \rho_1 & \theta_2 & & \\ & \ddots & \ddots & \\ & & \rho_{k-1} & \theta_k \\ & & & \rho_k \end{bmatrix} \in \mathbb{R}^{k\times k}, \tag{3.25}$$

where $\mathbf{P}_k \in \mathbb{R}^{n\times k}$, $\mathbf{Q}_k \in \mathbb{R}^{d\times k}$ and $\mathbf{P}_k^T\mathbf{P}_k = \mathbf{Q}_k^T\mathbf{Q}_k = \mathbf{I}_k$. The upper bidiagonal decomposition R_k is computed by using the Lanczos-like three term recurrence:

$$\begin{aligned} \mathbf{A}\mathbf{Q}_k &= \mathbf{P}_k\mathbf{R}_k \\ \mathbf{A}^T\mathbf{P}_k &= \mathbf{Q}_k\mathbf{R}_k^T + \theta_{k+1}\mathbf{q}^{k+1}\mathbf{e}_k^T \end{aligned} \Longrightarrow \begin{aligned} \mathbf{A}\mathbf{q}^1 &= \rho_1\mathbf{p}^1, \\ \mathbf{A}^T\mathbf{p}^j &= \rho_j\mathbf{q}^j + \theta_{j+1}\mathbf{q}^{j+1} \quad & j \le k, \\ \mathbf{A}\mathbf{q}^j &= \theta_j\mathbf{p}^{j-1} + \rho_j\mathbf{p}^j, \quad & j \le k, \end{aligned} \tag{3.26}$$

where θ_j's and ρ_j's are chosen so that $\|\mathbf{q}^j\|_2 = \|\mathbf{p}^j\|_2 = 1$, respectively. Note that $\mathbf{P}_k$ and $\mathbf{Q}_k$ are not needed to be orthogonal in `AAb_Solver`, therefore we do not need any reorthogonalization steps. Unlike the LSQR, we choose $\theta_1\mathbf{q}^1 = \mathbf{b}$ with $\theta_1 = \|\mathbf{b}\|_2$ so that the columns of the matrix $\mathbf{Q}_k$ constitute an orthonormal basis for the k-th order Krylov Subspace:

$$\operatorname{span}\{\mathbf{q}^1, \dots, \mathbf{q}^k\} = \mathcal{K}_k(\mathbf{A}^T\mathbf{A},\ \mathbf{b}) = \mathcal{K}_k(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I}_d,\ \mathbf{b}),\ \forall\mu \in \mathbb{R}_+.$$

Since the Krylov Subspaces is invariant under a constant shift, regularization does not affect this property. In the k-th iteration of the proposed `AAb_Solver`, let the solution estimate of the linear system be $\mathbf{x}^k = \mathbf{Q}_k\mathbf{y}^k$ for some vector $\mathbf{y}^k \in \mathbb{R}^k$, i.e., $\mathbf{x}^k \in \mathcal{K}_k(\mathbf{A}^T\mathbf{A},\ \mathbf{b})$, then we have:

$$(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_d)\mathbf{Q}_k\mathbf{y}^k = \mathbf{b}$$

which implies

$$\overline{\mathbf{R}}_k \mathbf{y}^k = \overline{\mathbf{R}}_k^{-T} \mathbf{Q}_k^T \mathbf{b} \overset{(a)}{=} \theta_1 \overline{\mathbf{R}}_k^{-T} \mathbf{e}_1,$$

where (a) is due to the choice of $\mathbf{q}^1$ and $\overline{\mathbf{R}}_k$ is obtained by applying a sequence of Givens rotation on $[\mathbf{R}_k^T \; \sqrt{\lambda}\mathbf{I}_k]^T$ in order to eliminate the sub-diagonal elements due to the regularization [80]. One instance of this elimination procedure is

$$\begin{bmatrix} \bar{\rho}_k & \theta_{k+1} \\ 0 & \rho_{k+1} \\ 0 & 0 \\ 0 & \sqrt{\lambda} \end{bmatrix} \rightarrow \begin{bmatrix} \bar{\rho}_k & c_k\theta_{k+1} \\ 0 & \rho_{k+1} \\ 0 & 0 \\ 0 & \overline{\lambda}_{k+1} \end{bmatrix} \rightarrow \begin{bmatrix} \bar{\rho}_k & \bar{\theta}_{k+1} \\ 0 & \bar{\rho}_{k+1} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \xrightarrow[\text{iteration}]{\text{next}} \begin{bmatrix} \bar{\rho}_{k+1} & \theta_{k+2} \\ 0 & \rho_{k+2} \\ 0 & 0 \\ 0 & \sqrt{\lambda} \end{bmatrix},$$

where $c_k = \rho_k / \bar{\rho}_k$, $s_k = \overline{\lambda}_k / \bar{\rho}_k$, $\bar{\theta}_{k+1} = c_k \theta_{k+1}$, $\overline{\lambda}_{k+1}^{\,2} = \lambda + (s_k\theta_{k+1})^2$ and $\bar{\rho}_{k+1} = \sqrt{\rho_{k+1}{}^2 + \overline{\lambda}_{k+1}^{\,2}}$. Since $\overline{\mathbf{R}}_k$ is an upper bidiagonal matrix, the inverse always exists and $\mathbf{f}^k := \overline{\mathbf{R}}_k^{-T} \mathbf{e}_1$ can be computed analytically as:

$$\phi_1 = \frac{\theta_1}{\bar{\rho}_1} \quad \text{and} \quad \phi_k = -\phi_{k-1} \frac{\bar{\theta}_k}{\bar{\rho}_k} \text{ where } \mathbf{f}^k = [\phi_1, \ldots, \phi_k]^T. \tag{3.27}$$

Furthermore, the solution at the k-th iteration, $\mathbf{x}^k = \mathbf{Q}_k \overline{\mathbf{R}}_k^{-1} \mathbf{f}^k$, can be obtain without computing any inversions by using the forward substitution. Define $\mathbf{D}_k = \mathbf{Q}_k \overline{\mathbf{R}}_k^{-1}$:

$$\left.\begin{array}{c} [\mathbf{D}_{k-1}, \; \mathbf{d}^k] \begin{bmatrix} \overline{\mathbf{R}}_{k-1} & \mathbf{e}_{k-1}\bar{\theta}_k \\ 0 & \bar{\rho}_k \end{bmatrix} = [\mathbf{Q}_{k-1}, \; \mathbf{q}^k] \\ \mathbf{D}_{k-1}\overline{\mathbf{R}}_{k-1} = \mathbf{Q}_{k-1} \\ \bar{\theta}_k \mathbf{d}^{k-1} + \bar{\rho}_k \mathbf{d}^k = \mathbf{q}^k \end{array}\right\} \begin{array}{l} \mathbf{d}^k = (\mathbf{q}^k - \bar{\theta}_k \mathbf{d}^{k-1})/\bar{\rho}_k \\ \mathbf{x}^k = \mathbf{x}^{k-1} + \phi_k \mathbf{d}^k, \end{array}$$

and the relative residual error that will be used as a stopping criterion can be found as:

$$\begin{aligned} \|\mathbf{A}^T\mathbf{A}\mathbf{x}^k + \lambda\mathbf{x}^k - \mathbf{b}\|_2^2 &= \|\mathbf{A}^T\mathbf{A}\mathbf{Q}_k\mathbf{y}^k + \lambda\mathbf{Q}_k\mathbf{y}^k - \mathbf{b}\|_2^2 = \|\mathbf{A}^T\mathbf{P}_k\overline{\mathbf{R}}_k\mathbf{y}^k - \mathbf{b}\|_2^2 \\ &= \| \left(\mathbf{Q}_k\overline{\mathbf{R}}_k^T + \theta_{k+1}\mathbf{q}^{k+1}\mathbf{e}_k^T\right) \overline{\mathbf{R}}_k\mathbf{y}^k - \mathbf{b}\|_2^2 \\ &\overset{(i)}{=} \|\overline{\mathbf{R}}_k^T\overline{\mathbf{R}}_k\mathbf{y}^k - \mathbf{Q}_k^T\mathbf{b}\|_2^2 + \|\theta_{k+1}\mathbf{q}^{k+1}\mathbf{q}_k^T\overline{\mathbf{R}}_k\mathbf{y}^k - \left(\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^T\right)\mathbf{b}\|_2^2 \\ &= \left|\phi_k\bar{\theta}_{k+1}\right| = |\phi_{k+1}\bar{\rho}_{k+1}|. \end{aligned}$$

The first norm in (i) is zero since the linear system is always consistent. The second term in the second norm is also zero, since $\mathbf{b} \in \text{span}(\mathbf{Q}_k)$ by the initial choice of $\theta_1 \mathbf{q}^1 = \mathbf{b}$. By definition, $\mathbf{f}^k = \overline{\mathbf{R}}_k \mathbf{y}^k$ gives the final results. The overall algorithm is given in Algorithm 7. The `AAb_Solver` is also a Krylov Subspace method, therefore, it finds the solution in at most $\min(n, d, m)$ iterations in the exact arithmetic, but far fewer number of iterations is sufficient for our purpose.

Algorithm 7 `AAb_Solver` (for problems in the form of $(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})x = \mathbf{b}$)

```
Input: $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b}, \lambda, \epsilon$                                   complexity
▷ choose $\rho$'s and $\theta$'s to make $\|\mathbf{p}\|_2 = \|\mathbf{q}\|_2 = 1$
$\theta_1 \mathbf{q} = \mathbf{b}$                                                                  $3n$
$\rho \mathbf{p} = \mathbf{A}\mathbf{v}$                                                            $(m+3)n$

$\bar{\rho} = \sqrt{\rho^2 + \lambda}, \quad c = \rho/\bar{\rho}, \quad s = \sqrt{\lambda/\bar{\rho}}, \quad \phi = \theta_1/\bar{\rho}$    $O(1)$
$\mathbf{d} = \mathbf{q}/\bar{\rho}$                                                                $n$
$\mathbf{x} = \phi \mathbf{d}$                                                                      $n$
while $t \geq \epsilon$ do
    $\theta \mathbf{q} := \mathbf{A}^T \mathbf{p} - \rho \mathbf{q}$                                $(m+5)n$
    $\rho \mathbf{p} := \mathbf{A}\mathbf{q} - \theta \mathbf{p}$                                   $m(n+5)$

    $\bar{\lambda}^2 := \lambda + (s\theta)^2, \quad \bar{\theta} = c\theta$                        $O(1)$
      $\bar{\rho} := \sqrt{\rho^2 + \bar{\lambda}^2}, \quad c = \rho/\bar{\rho}, \quad s = \bar{\lambda}/\bar{\rho}$    $O(1)$

      $\mathbf{d} := (\mathbf{q} - \bar{\theta}\mathbf{d})/\bar{\rho}$                            $3m$
      $\phi := -\phi\bar{\theta}/\bar{\rho}$                                                        $O(1)$
      $\mathbf{x} := \mathbf{x} + \phi \mathbf{d}$                                                  $2n$
        $t = |\phi\bar{\rho}|/\theta_1$                                                             $O(1)$
end while
```

Efficient solutions for linear systems in the form of $(\mathbf{A} + \lambda\mathbf{I})x = \mathbf{b}$ for a symmetric matrix or $(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})\mathbf{x} = \mathbf{b}$ for a rectangular matrix have been well studied subject. In the first case, Lanczos tridiagonalization algorithm can be used for deriving a stable solver [81]. In the second case, which is our main concern, if the lower bidiagonalization processes (*bidiag1* in [49]) is used such as in [82], then a tridiagonal system in the form of $\mathbf{B}_k^T \mathbf{B}_k \mathbf{y}^k = \theta_1 \mathbf{e}_1$ must be solved where $\mathbf{B}_k \in \mathbb{R}^{k+1 \times k}$ is a lower bidiagonal matrix. This system can be solved by first eliminating the lower diagonal elements in the tridiagonal matrix $\mathbf{B}_k^T \mathbf{B}_k$ and then by using forward substitution. However, the condition number of $\mathbf{B}_k^T \mathbf{B}_k$ is the square of the condition number of $\mathbf{B}_k$ and thus increases the instability of the operations in the inexact arithmetic. Therefore, in the proposed `AAb_Solver`, we use upper bidiagonalization process to solve a tridiagonal system in the form of $\mathbf{R}_k^T \mathbf{R}_k \mathbf{y}^k = \theta_1 \mathbf{e}_1$. The major advantage of this form over the one obtained by lower bidiagonal matrix $\mathbf{B}_k$ is that $\mathbf{R}_k^{-T} \mathbf{e}_1$ can be calculated analytically as in eq. (3.27). Then the solution $\mathbf{y}^k$ can be obtained via backward substitution. In this way, we avoid both squaring the condition number and the elimination process of the lower diagonal entries. As a result, we obtain a solver with better stability properties and with slightly lower computational requirements.

3.4 Numerical Experiments and Comparisons

We compare the operation counts required by the algorithms to obtain a certain level of accuracy in the solution approximation metric. For a fair comparison, we have implemented all the proposed algorithms in this manuscript as well as those that are used for the comparisons in MATLAB which can be found in the following link: `https://github.com/ibrahimkurban/M-IHS`.

3.4.1 Experiment setups

The coefficient matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ was generated for various sizes as follows: we first sampled the entries of $\mathbf{A}$ from the distribution $\mathcal{N}(1_d, \mathbf{\Gamma})$ where $\Gamma_{ij} = 5 \cdot 0.9^{|i-j|}$ so that the columns are highly correlated with each other. Then by using the SVD, we replaced the singular values with *philips* profile provided in RegTool [41]. Unless indicated otherwise, we scaled the singular values to set the condition number $\kappa(\mathbf{A})$ to 10^8 and we used the same input signal provided by RegTool. In this way, we have obtained a challenging setup for any first order iterative solvers to compare their performances. In all the experiments, the same setup has been used unless indicated. We counted the number of operations according to Hunger's report [83]. All the reported results have been obtained by averaging over 32 MC simulations.

3.4.2 Compared methods and their implementation details

In the conducted comparison study we used a total of 7 previously proposed techniques that can be briefly described as follows. The Accelerated Randomized Kaczmarz (ARK) uses Nesterov's method for accelerating the randomized Kaczmarz algorithm [84, 85]. The CGLS is an adaptation of the CG for the LS problems [18]. The rest of the techniques are the state of the art randomized preconditioning techniques which can reach any level of desired accuracy within a bounded number of iterations. The Blendenpik uses the **R** matrix in the QR decomposition of the sketched matrix **SA** as the preconditioning matrix for the LSQR algorithm just like the method proposed by Rokhlin et al. [56, 57] and it uses Randomized Orthonormal System (ROS) to generate the sketched matrix [14]. The LSRN uses the **V** matrix in the SVD similar to the Blendenpik. In spite of its high running complexity, for parallelization purposes, the Gaussian sketch matrices are preferred in the LSRN. In addition to the LSQR, also the CS can be preferred in the LSRN as the core solver in distributed computational environments [29]. The IHS uses the sketched Hessian as the preconditioning matrix for the Gradient Descent. The Accelerated IHS (A-IHS) uses this idea for the CG algorithm in over-determined problems. The dual counter-part of the A-IHS algorithm, A-IDRP, is shown to be faster than the Dual Random Projection algorithm proposed in [86], so we did not include the DRP in the simulations. Additionally, we include a CS variant of the IHS (IHS-CS) to the comparisons: we combined the randomized preconditioning idea of the IHS with the preconditioned CS method [13]. We found the bounds for the eigenvalues in the same way as in the LSRN. We have solved the low dimensional sub-problems required by all the IHS variants by taking the QR decomposition, but for *inexact* schemes, we have used the proposed `AAb_Solver` with a constant forcing term. Although the inexact approach is also applicable for the accelerated algorithms proposed in [1], we did not include them in the simulations since their exact versions are outperformed by the *Exact* `M-IHS` variants in all settings. Except for the LSRN variants which use Gaussian sketch matrices, we used Discrete Cosine Transform in the ROS for all the compared techniques.

3.4.3 Results obtained for un-regularized LS problems

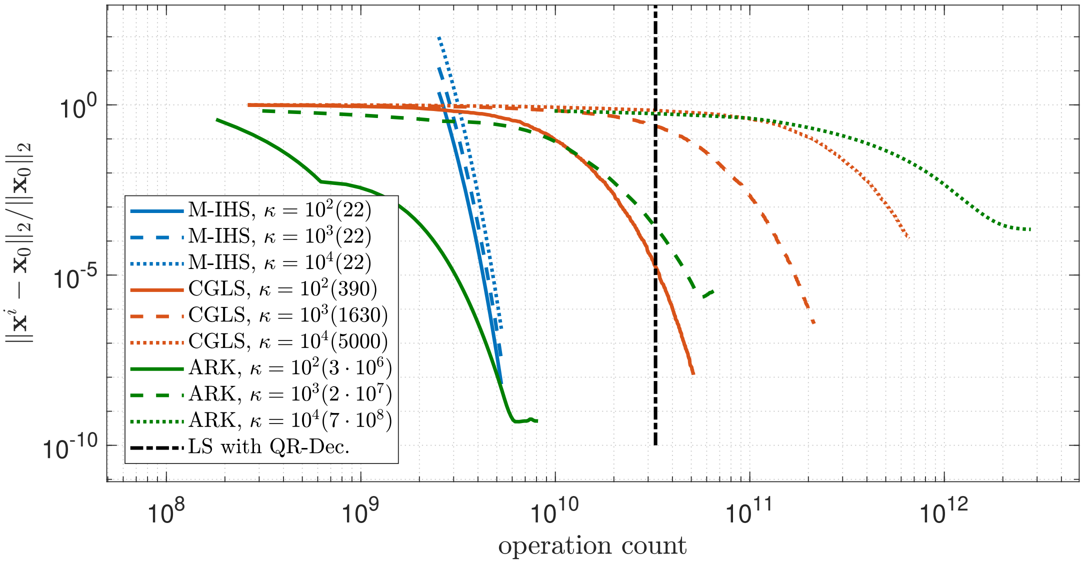


Figure 3.3: *Performance comparison of the `M-IHS`, ARK and CGLS on an un-regularized LS problem with size* $2^{16} \times 500$.

In the first experiment, we did not include noise in the linear system to emphasize the convergence rate that the algorithms can provide in such severely ill posed problems. To make the problem more challenging, for this experiment only we sampled the input vector $\mathbf{x}_0$ from uniform distribution $\text{Uni}(-1, 1)$. In such scenarios, convergence rates of Krylov subspace-based iterative solvers without preconditioning fall to its minimum value since the energy of the input is distributed equally over the range space of $\mathbf{A}$.

In Figure 3.3, the `M-IHS` is compared with the ARK and the CGLS on a highly over-determined problem for a set of different condition numbers. When the condition number κ of the coefficient matrix $\mathbf{A}$ increases, convergences of the CGLS and the ARK degrade considerably while the performance of the `M-IHS` technique remains unaffected. Although the ARK performs better than the CG for low κ values, its convergence is influenced worse than the CG by the increasing condition number. The `M-IHS` requires substantially less operations than the ARK and the CGLS that is unable to converge even in d iterations due to round-off errors.

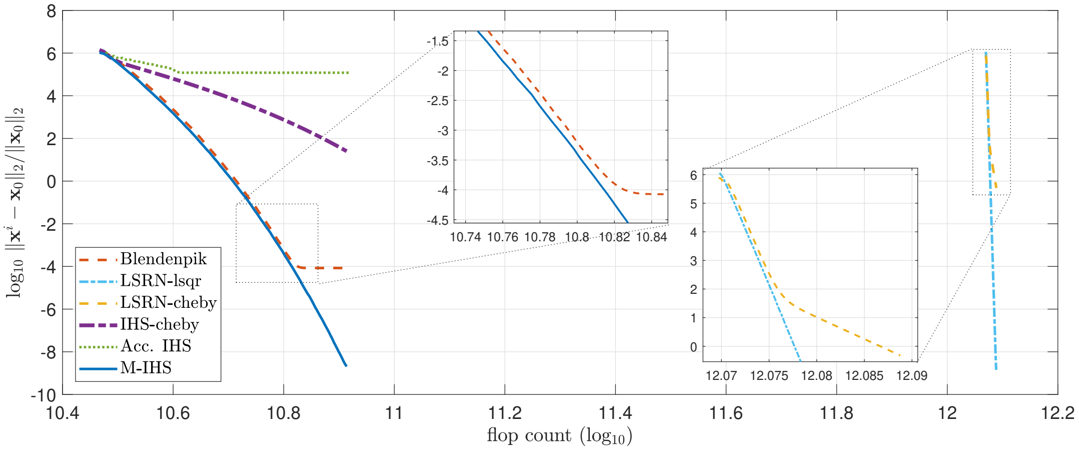


Figure 3.4: *Performance comparison on an un-regularized LS problem with size* $2^{16} \times 2000$*. In order to compare the convergence rates, number of iterations for all solvers are set to* $N = 100$ *with the same sketch size:* $m = 4000$*. According to the Corollary 3.2.2.1, we expect the* `M-IHS` *to reach an accuracy:* $\left\|\mathbf{x}^N - \mathbf{x}_0\right\|_2 \leq \kappa(\mathbf{A}) \left\|\mathbf{x}_0\right\|_2 \left(1/\sqrt{2}\right)^N = 9 \cdot 10^{-8}$*, which closely fits to the observed case.*

In Figure 3.4, the `M-IHS` is compared with the randomized preconditioning techniques described above. Due to high running time of the Gaussian sketches, $O(mnd)$, the LSRN variants require more operations (for the size of the problems considered here approximately 10 times larger) than the others. Due to the lack of inner product calculations, the `M-IHS` requires slightly fewer operations than the Blendenpik, nonetheless, it reaches to the same accuracy with the LSRN-LSQR. The A-IHS algorithm has the worst performance which is expected in the un-regularized problems, since it is adapted on the CG technique that can be unstable for the un-regularized LS problems due to the high condition number [49]. The convergence of the CS-based techniques, both of the IHS and the LSRN variants, are substantially slower than the `M-IHS`, which suggests that the `M-IHS` algorithm can take the CS's place in those applications where parallel computation is an option.

3.4.4 Results obtained for regularized LS problems

We tested robustness of the methods against noise on regularized LS problems by using an additive i.i.d. Gaussian noise at level $\|\mathbf{w}\|_2 \big/ \|\mathbf{A}\mathbf{x}_0\|_2 = 1\%$. For this purpose, the optimal regularization parameter that minimizes the error $\|\mathbf{x}(\lambda) - \mathbf{x}\|_2$ is provided to all techniques. Each technique is allowed to conduct a total of 20 iterations. Results for strongly over-determined and strongly under-determined cases can be seen in Figure 3.5 and Figure 3.6, respectively. We used a sketch size of $m = \min(n, d)$ to emphasize the promise of the RP techniques although such sizes are not applicable for the LSRN variants. Even if the sketch size has been increased further, the convergence of the LSRN variants were considerably slower than the others; so we leave out the LSRN variants from the comparison set in the regularized settings. Also, in the regularized setup, the A-IHS and A-IDRP methods are slower than the Blendenpik, IHS-CS and `M-IHS` variants. Besides, the inexact schemes proposed for the `M-IHS` and `Dual M-IHS` require significantly less operations to reach to the same level of accuracy as their exact versions. Although the inexact schemes require approximately 10 times less operations then their exact versions in these setups; the saving gets larger as the sketch size increases as examined in Section 3.4.5, because while any full decomposition requires $O(mr^2)$ operations, approximately solving the sub-problem requires only $O(mr)$ operations.

As long as the statistical dimension of the problem is small with respect to the dimensions of coefficient matrix $\mathbf{A}$, Lemma 3.2.2 implies eligibility of sketch sizes that are smaller than the rank, $m \leq \min(n, d)$. This implication can be verified in Figure 3.7 on which we showed the performance of the `Primal Dual M-IHS` techniques. Here, the *inexact* schemes of the `M-IHS` and `Dual M-IHS` use a sketch size $m = 2 \cdot \mathsf{sd}_\lambda(\mathbf{A})$. The primal dual schemes use $m_1 = m_2 = 2 \cdot \mathsf{sd}_\lambda(\mathbf{A})$ except for the `Primal Dual M-IHS` shown as a green curve which uses $m_1 = m_2 = 8 \cdot \mathsf{sd}_\lambda(\mathbf{A})$. All the methods are allowed to conduct $N = 60$ iterations except the `Primal Dual M-IHS` with larger sketch size is allowed to conduct only 20 iterations. The number of inner iterations are restricted by $M = 25$ for all the primal dual schemes. Lastly, a fixed forcing term $\epsilon_{sub} = 0.1$ is used in the `AAb_Solver` for

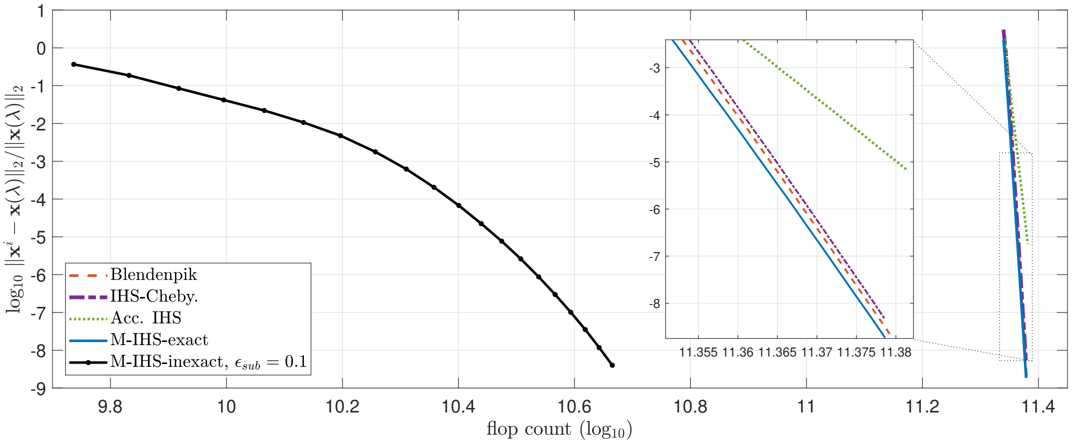


Figure 3.5: *Performance comparison on a regularized LS problem ($n \gg d$) with dimensions* $(n, d, m, \mathsf{sd}_\lambda(\mathbf{A})) = (2^{16}, 4000, 4000, 443)$. *According to the Corollary 3.2.2.1,* `M-IHS` *is expected to satisfy:* $\left\|\mathbf{x}^N - \mathbf{x}_0\right\|_2 \leq \|\mathbf{x}_0\|_2 \sqrt{\kappa(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_d)} \left(\sqrt{443/4000}\right)^N = 6 \cdot 10^{-9}$ *which is almost exactly the case. The Inexact* `M-IHS` *requires significantly fewer operations to reach the same accuracy as others. For example to obtain an ($\eta = 10^{-4}$)-optimal solution approximation, the Inexact* `M-IHS` *requires approximately 10 times less operations than any techniques that need factorization or inversion of the sketched matrix.*

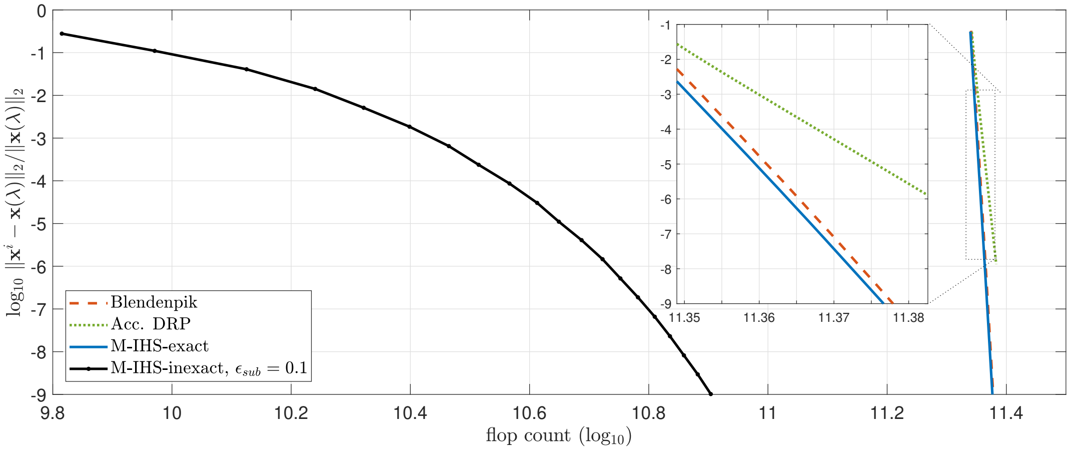
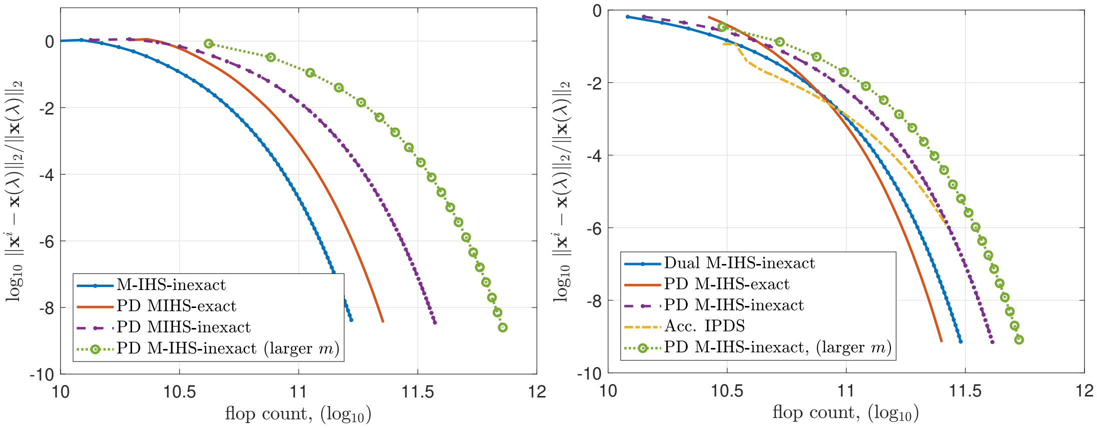


Figure 3.6: *Performance comparison on a regularized LS problem ($n \ll d$) with dimensions* $(n, d, m, \mathsf{sd}_\lambda(\mathbf{A})) = (4000, 2^{16}, 4000, 462)$. *The comments in Figure 3.5 are also valid for this case. The Inexact scheme for* `Dual M-IHS` *is capable of significantly reducing the complexity.*



(a) $n \geq d$ *and* $\mathsf{sd}_\lambda(A) = 680$

(b) $n \leq d$ *and* $\mathsf{sd}_\lambda(A) = 825$

Figure 3.7: *Performance comparison on a regularized LS problems with square-like dimensions. The problem dimensions are set to* $\max(n, d) = 5 \cdot 10^4$ *and* $\min(n, d) = 10^4$ *with a noise level of* 10%. *The results verifies two hypotheses: first the sketch size for the* `M-IHS` *variants can be chosen proportional to the statistical dimension even if it becomes smaller than the size of the coefficient matrix. Second, the coefficient matrix can be sketched from both sides to reduce computational complexity.*

all the inexact schemes. Applying a second dimension reduction may not seem to create significant computational saving, but this approach produces smaller sub-problems than the `M-IHS` and the `Dual M-IHS` techniques therefore enables estimation of parameters such as λ with far fewer number of operations. Lastly, the `Primal Dual M-IHS` variants have a noticeably higher rate of convergence than the A-IPDS algorithm which is based on the CG technique.

3.4.5 Scalability to larger size problems

In this section, as the size of the coefficient matrix and the sketch size increase we show that the saving gained by the inexact schemes become critically more important. For this purpose, the algorithms were run on the over-determined problems with size $5 \cdot 10^4 \times \gamma \cdot 500$ where $\gamma \in \{1,\ 2,\ 4,\ 8,\ 16\}$. The sketch size was chosen as $m = d = \gamma \cdot 500$ and the regularization parameter was set to 0.1453 for all the experiments so that $\mathsf{sd}_\lambda(\mathbf{A}) = d/10$ remains the same for all the experiments. The data was generated by using the setup described in

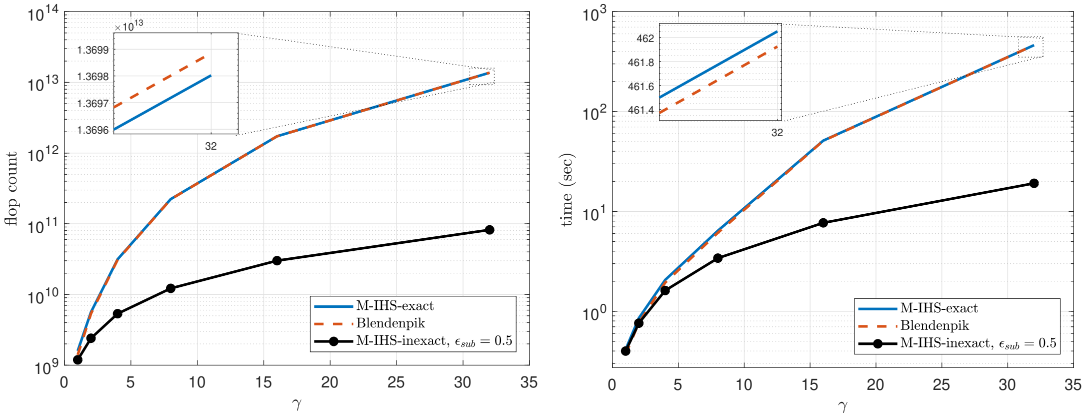


Figure 3.8: *Complexity of the algorithms in terms of operation count and computation time on a set of* $5 \cdot 10^4 \times 500 \cdot \gamma$ *dimensional over-determined problems with* $m = d$ *and* $\mathsf{sd}_\lambda(\mathbf{A}) = d/10$.

Section 3.4.1. Note that the convergence properties of the proposed techniques depend only on the statistical dimension but not directly to the decay rate of the singular values. To show this, for these experiments, we used *heat* singular value profile that has significantly lower decay rate than the *philips* profile used earlier. The experiments were realized on a desktop with 4Ghz i7-4790K CPU processor and 32Gb RAM. The flop count and wall clock time of the algorithms to reach to an ($\eta = 10^{-4}$)-optimal solution approximation are shown in Figure 3.8. As d and m reach thousands, the number of operations required by the exact schemes (Blendenpik and `M-IHS`) becomes larger than 100 times of the operation count required by the inexact scheme. Moreover, the exact schemes need 25 time longer time than the inexact scheme to reach the desired accuracy. Additionally, the operation counts and elapsed time in each stage of the algorithms can be seen in Figure 3.9 which shows that even the cost of the decomposition applied on the sketched matrices reaches to prohibitive levels for large scale problems. Hence the use of solvers such as `M-IHS` variants that allow inexact schemes is the only practical choice in these regimes. In these experiments, for the estimation of the statistical dimension, we set $T = 2$ and $\epsilon_{tr} = 0.5$. The additional cost of the $\mathsf{sd}_\lambda(\mathbf{A})$ estimation for the proposed `M-IHS` variants becomes negligibly small when $\mathbf{R}$-factor is utilized; for the inexact schemes, still it has a low cost, around the cost of one M-IHS-inexact iteration, that does not cause an issue unlike a matrix decomposition.

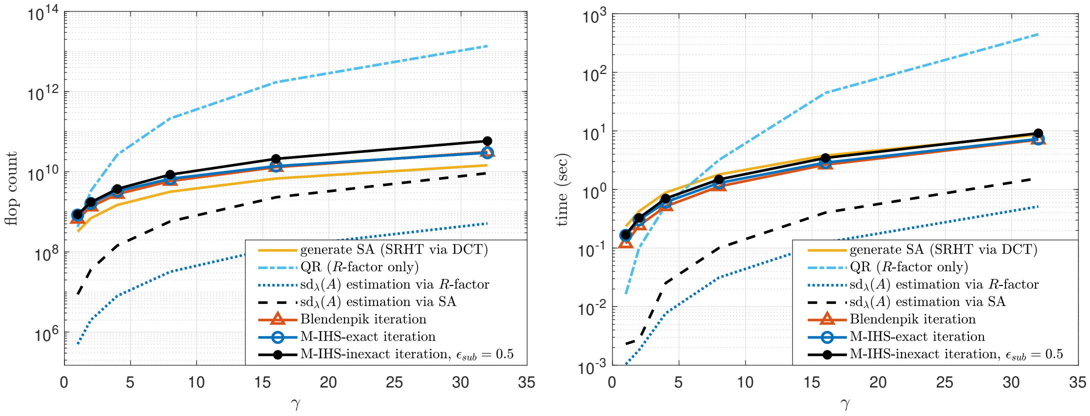


Figure 3.9: *Complexity of the each stage in terms of operation count and computation time on a set of* $5 \cdot 10^4 \times 500 \cdot \gamma$ *dimensional over-determined problems with* $m = d$ *and* $\mathsf{sd}_\lambda(\mathbf{A}) = d/10$*. All methods contain* **SA** *generation stage. The Blendenpik and* `M-IHS`*-exact contain also QR decomposition stage but* `M-IHS`*-inexact does not. The* `M-IHS`*-exact estimates* $\mathsf{sd}_\lambda(\mathbf{A})$ *by using the* **R***-factor while the* `M-IHS`*-inexact uses directly* **SA** *matrix and the* `AAb_Solver` *as proposed in Algorithm 11. The results show that the matrix decompositions are the main computational bottleneck for the exact schemes in large scale problems where the advantage of the inexact schemes becomes more significant.*

Remark 4. *Benchmarking of the exact and inexact schemes by using wall clock time in MATLAB is not a fair comparison because for-loops in the interpreted languages such as MATLAB is well known to be much slower than the loops in compiled languages such as C. Most of the decompositions in MATLAB have C-based implementation with professional use of BLAS operations, while the inexact schemes are based on a for-loop. Therefore, we prefer to rely on the operation counts. However, to give an opinion, in spite of the disadvantages we demonstrate timing as well.*

3.4.6 Effect of the statistical dimension on the performance of the inexact schemes

The inexact schemes become more efficient as the statistical dimension decreases since the sub-problems are solved in less iterations. To show the effect of varying statistical dimension on the complexity of the algorithms, we used over-determined problems with size $5 \cdot 10^4 \times 16 \cdot 10^3$ and varied the

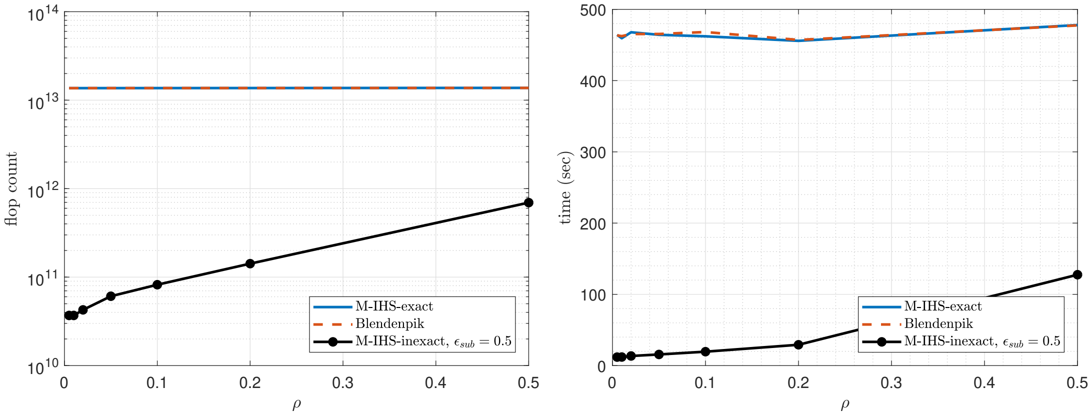


Figure 3.10: *Complexity of the algorithms in terms of operation count and computation time on a $5 \cdot 10^4 \times 4 \cdot 10^3$ dimensional problem for different $\rho = \mathsf{sd}_\lambda(A)/d$ ratios.*

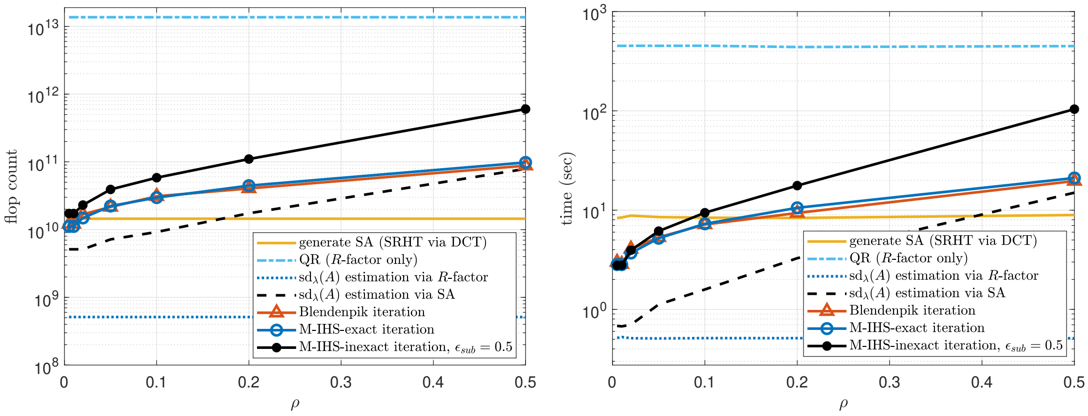


Figure 3.11: *Complexity of each stage in terms of operation count and computation time on a $5 \cdot 10^4 \times 4 \cdot 10^3$ dimensional problem for different $\rho = \mathsf{sd}_\lambda(\mathbf{A})/d$ ratios. Stages of each algorithm is given in Figure 3.9.*

regularization parameter to obtain different $\rho = \mathsf{sd}_\lambda(\mathbf{A})/d$ ratios where $\rho \in \{0.5\%,\ 1\%,\ 2\%,\ 5\%,\ 10\%,\ 20\%,\ 50\%\}$. The sketch size was chosen as $m = d$ and *heat* profile was used. As in Section 3.4.5, the flop count and wall clock time of the algorithms to reach to an ($\eta = 10^{-4}$)-optimal solution approximation for the problems with different statistical dimensions are shown in Figure 3.10. Complexity of the exact schemes increases by the increasing statistical dimension since convergence rate $\mathsf{sd}_\lambda(\mathbf{A})/m$ decreases as m remains constant. Complexity of the inexact scheme increase faster since sub-problems require more iterations as the effective range space gets larger. The effect of the increasing statistical dimension over the different stages of the algorithms are shown in Figure 3.11.

For the estimation of $\mathtt{sd}_\lambda(\mathbf{A})$, same parameters $T = 2$ and $\epsilon_{tr} = 0.5$ were used as early. Even for the large ρ ratios, utilizing a sub-solver is still more effective than computing a matrix decomposition.

3.5 Contributions and Conclusion

In this chapter, we proposed a group of RP-based iterative solvers for large scale LS problems. As shown by detailed analyses of their convergence behaviour, the proposed `M-IHS` variants can be used for any dimension regimes with significant computational savings if the statistical dimension of the problem is sufficiently smaller than at least one dimension of the coefficient matrix. Our guarantees, presented in Theorem 3.2.1 and Corollary 3.2.1.1, are based on the solution approximation metric given in eq. (2.14) as opposed to the results obtained for cost approximation metric given in eq. (2.13). In Lemma 3.2.2, we improved the known lower bounds on the sketch size of various randomized distribution for obtaining a pre-determined convergence rate with a constant probability. These guarantees can be readily extended to any other sketching types by using the AMM property defined in [69]. When tighter bounds for the AMM property will be available in the future, the bounds derived in this work can be automatically improved as well. Although our bounds for the dense sketch matrices such as Subgaussian or Randomized Orthonormal Systems (ROS) are the same as in [36], we gained slightly better results for the sparse sketch matrices. Additionally, we provide some empirical bounds for the sketch size and the rate of convergence in Corollary 3.2.2.1 which is remarkably tight as demonstrated through numerical experiments.

In Algorithm 7, we extend the idea of LSQR into the linear problems in the form of $A^TAx = b$ which we need to solve during the iterations of all proposed *Inexact* `M-IHS` variants and of the Newton Sketch [25]. Similar to the stability advantage of the LSQR over the CGLS technique [49], the proposed sub-solver solves the system in the above form without squaring the condition number as opposed to the techniques such as the symmetric CG and the symmetric Lanczos procedures.

The main advantage of the proposed `M-IHS` variants over the state of the art randomized preconditioning techniques such as the Blendenpik, A-IHS and LSRN is their ability to use inexact schemes that avoids matrix decompositions or inversions. As demonstrated in a wide range of numerical experiments, computational saving provided by the proposed solver becomes decidedly significant in large scale problems. Lastly, the proposed `M-IHS` variants avoid using any inner products in their iterations and they are shown to be faster than CS-based randomized preconditioning algorithms. Therefore the proposed `M-IHS` variants are strong candidates to be the techniques of choice in parallel or distributed architectures.

Chapter 4

Proposed Hybrid M-IHS Techniques

In this chapter, after examining the relation between the regularization of the full problem given in eq. (2.3) and the projected problem given in eq. (2.10), the computational bottlenecks in the conventional hybrid methods are discussed in Section 4.1. Then to remedy these bottlenecks, the `Hybrid-M-IHS` techniques are proposed in Section 4.2. In Section 4.3, the proposed `M-IHS` techniques are compared with the conventional hybrid methods and the direct methods on several realistic problems such as image de-blurring and tomography at various SNR levels. The chapter is ended in Section 4.4 by stating the contributions and conclusion remarks.

4.1 Computational Bottlenecks in the Conventional Hybrid Methods

In Krylov Subspace methods such as the LSQR, solution to the full problem at the k-th iteration is obtained by the transformation $\mathbf{x}^k(\lambda) = \mathbf{Q}_k \mathbf{y}^k(\lambda)$ where $\mathbf{Q}_k$ is constructed along with the GKL procedure and its columns constitute an orthonormal basis for the k-th order Krylov Subspace [10]. As a result, we have the following equality

$$\left\|\mathbf{b} - \mathbf{A}\mathbf{x}^k(\lambda)\right\|_2 = \left\|\beta_1 \mathbf{e}_1 - \mathbf{B}_k \mathbf{y}^k(\lambda)\right\|_2, \tag{4.1}$$

between the two problems in eq. (2.3) and eq. (2.10) which suggests that the numerators of the full GCV function in eq. (2.8) and the projected version in eq. (2.11) are exactly the same. Hence, the difference between the two estimators the G_{full} and the G_{proj} is because of the differences in their respective denominators, i.e., different degrees of freedom estimated for the same residual error. To understand the cause of this difference, consider the spectrum of $\mathbf{B}_k$ and $\mathbf{A}$. For some (k_1, k_2) integer pair such that $k_1 \leq k \leq k_2 \leq r$, the first k_1 singular values of $\mathbf{B}_k$ are known to be very good approximations to the first k_1 singular values of $\mathbf{A}$, but the rest of the singular values of $\mathbf{B}_k$ approximates the singular value spectrum between σ_{k_1} and σ_{k_2} of $\mathbf{A}$ [7, 87]. Assume $k^* \leq k_1$ so that the inaccurate small singular values in $\mathbf{B}_k$ do not affect the filtering coefficients ϕ_i's that are close to 1 and consequently $\mathsf{sd}_\lambda(\mathbf{A}) \approx \mathsf{sd}_\lambda(\mathbf{B}_k)$, which is equivalent to assuming $\phi_i \approx 0$ for $i \geq k$ as in [40]. Even for such cases, the denominator of G_{proj}:

$$\mathsf{tr}\left(\mathbf{I}_{k+1} - P_{\mathbf{B}_k}(\lambda)\right) = k + 1 - \mathsf{sd}_\lambda(\mathbf{B}_k) \approx k + 1 - \mathsf{sd}_\lambda(\mathbf{A}),$$

is much smaller than the actual degrees of freedom which is accurately estimated by the denominator of G_{full} function as $n - \mathsf{sd}_\lambda(\mathbf{A})$. This error in G_{proj} can be reduced by using the following modification as demonstrated in fig. 4.1.

$$G_{modified}(\lambda) = \frac{\left\|\beta_1 \mathbf{e}_1 - \mathbf{B}_k \mathbf{y}^k(\lambda)\right\|_2}{n - \mathsf{tr}\left(P_{\mathbf{B}_k}(\lambda)\right)}. \tag{4.2}$$

Remark 5. *The minimizer of $G_{modified}$ in eq.* (4.2) *is equal to the minimizer of the W-GCV in eq.* (2.12) *for $\omega_k = \frac{1+k}{n}$ which is proposed in [40]. This equivalence can be seen by taking the derivative of both functions in eq.* (4.2) *and eq.* (2.12) *with respect to λ. Roots of the resulting polynomials are the same, hence equating them to zero produce the same solutions.*

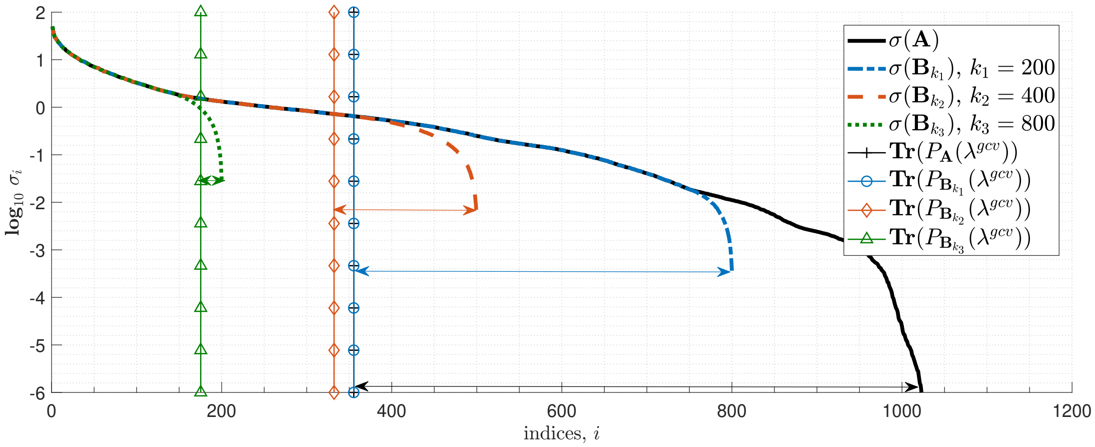


Figure 4.1: *The main difference between the projected and the naive GCV functions: the lengths of the horizontal lines with double arrows are equal to the denominator of the GCV functions applied on the projected problems with different sizes. Here, λ is set to λ^{gcv} for all cases so that the numerators of the respective GCV functions are the same. As demonstrated by the length of the respective horizontal lines, the degrees of freedom in the residual error are erroneously determined by all of the projected estimators. The correct value, $n - sd(A)$, can be obtained by the proposed correction if the projected problem with size k_3 is used, since $\mathsf{sd}_\lambda(B_{k_3}) \approx \mathsf{sd}_\lambda(A)$ as seen on the overlapped vertical lines with $\circ$ and $+$ shaped markers.*

The above analysis suggest that the regularization parameter of the full problem can be estimated from the projected problems that is obtained in the iterations of the GKL procedure by using the estimator $G_{modified}$ given in eq. (4.2) as long as the size of the bidiagonal matrix and therefore the number of the iteration of the hybrid method is sufficiently larger than the effective rank of the problem as demonstrated in Figure 4.1. If the number of iterations k is not sufficiently large to satisfy the condition $k_1 \geq k^*$, then the regularization parameter estimated from the projected problem might not serve as a proper estimate for the full problem. In [10, 47, 51], the GCV is proposed to be used for determining the number of iterations as well, but as shown in Section 3.4, the GCV criterion frequently terminates the iterations much earlier than the effective rank k^*, which results in excessive overestimation of the regularization parameter and excessively smoothed reconstructions of the solution.

When a proper regularization parameter for the full problem is sought, the computational complexity of the conventional hybrid methods become $O(ndk^* +$

$(n + d + k^*)(k^*)^2)$ where the first term is due to the matrix-vector multiplication with coefficient matrix $\mathbf{A}$ during the GKL process, the second and the third term are due to the re-orthogonalization steps [40, 46, 50] and the last term is due to the change of basis. The first term of the complexity expression is the source of a significant challenge in the distributed memory environments where even a few hundreds of distributed matrix-vector computations will not be tolerable due to significant increase in the overall computation time. In such applications, re-orthogonalization is another bottleneck since each memory node in the network stores only a certain partition of the left and the right factors constructed in the bidiagonal decomposition. For instance in highly over or underdetermined problems, if full re-orthogonalization with classical Gram-Schmidt procedure is used, then each iteration of the conventional hybrid methods requires at least three distributed matrix-vector computations given that the memory space of the master/central node is $O(\min(n, d)^2)$. In decentralized networks or square-like regimes where n and d scale similar, analyses of the algorithms become too complicated for discussing here due to the limited space. The GCV given in eq. (2.11) or eq. (4.2) can be minimized by utilizing the bidiagonal structure of $\mathbf{B}_k$ matrix with less than $O((k^*)^2)$ operations so its complexity is omitted. If it is minimized by using the SVD, then an additional cost of $O((k^*)^4)$ arises (see implementations in [11, 47, 51]). Even if the SVD is not computed in every iteration, it is needed more than one times to detect a viable regularization parameter but we have not encountered with any strategy in the literature to keep the number of times that the SVD is computed as fewer as possible.

4.2 Derivation and Analyses of the Proposed Hybrid M-IHS Techniques

To reduce the number of multiplications with the coefficient matrix and hence the overall time complexity, here we propose a group of RP-based hybrid schemes that obtain reliable estimates for the regularization parameter from the lower dimensional sub-problems that arise during the iterations of the `M-IHS` variants [37].

In the following sections, detailed derivations of the proposed hybrid techniques for each M-IHS variant are presented. For the sake of simplicity, we will use the SVD only in the analyses, but in Appendix C, we provide the GKL procedure-based algorithms that need only level-1 and level-2 BLAS operations [88]. Also, in the numerical results presented in Section 3.4, GKL procedure-based versions of the proposed techniques are used.

4.2.1 Hybrid M-IHS for highly over-determined problems

The proposed Hybrid M-IHS uses the following update at the i^{th} iteration:

$$\begin{aligned} \big((\mathbf{SA})^T(\mathbf{SA}) + \lambda_i \mathbf{I}_d\big)\, \Delta\mathbf{x}^i(\lambda_i) &= \mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i) - \lambda_i \mathbf{x}^i \\ \mathbf{x}^{i+1} &= \mathbf{x}^i + \alpha_i \Delta\mathbf{x}^i(\lambda_i) + \beta_i(\mathbf{x}^i - \mathbf{x}^{i-1}), \end{aligned} \tag{4.3}$$

where λ_i is the regularization parameter and $\Delta\mathbf{x}^i(\lambda_i)$ denotes the HS step that will be taken to compute the $(i+1)^{th}$ iterate. The difference from the naive M-IHS solver proposed in [37] is due to variable λ_i, α_i and β_i parameters. If the regularization parameter λ were known, the naive M-IHS would approximate the Hessian matrix in the Newton method to gain considerable saving in the computation and estimate the fixed momentum parameters that maximize the rate of convergence. In the absence of such prior information on λ, we aim to obtain an estimate of the regularization parameter λ_i that will be used in the update eq. (4.3) and to adjust the momentum parameters accordingly. For this purpose, $\mathbf{x}(\lambda)$ in the GCV function given in eq. (2.8) can be replaced with $\mathbf{x}^i + \Delta\mathbf{x}^i(\lambda)$ as proposed in [89]. However in this case, for each λ used in the numerical minimization of the GCV function, it is required to access to the full coefficient matrix, which becomes infeasible for large scale problems in the distributed computational environments. To avoid this computational bottleneck, we will use the GCV formulation given in eq. (2.9) and utilize only the residual error projected on the range space of $\mathbf{A}$ for the parameter estimation. Consider the following identity due to the first order optimality condition of the problem given in eq. (2.3):

$$\lambda \mathbf{A}^{\ddagger}\mathbf{x}(\lambda) = \mathbf{U}^T(\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)), \tag{4.4}$$

where the right hand side (RHS) is the numerator of the risk function in eq. (2.9) when k is set to d, and $\mathbf{A}^{\ddagger}$ denotes the Moore-Penrose inverse $(\mathbf{A}^T)^{\dagger}$. Equation (4.4) means that the residual error projected onto the range space of $\mathbf{A}$ can be computed by using the regularized solution itself but still $\mathbf{A}$ is needed. To reduce the complexity of the inversion and the number of access to the full data, if the pseudo-inverse of $\mathbf{A}^T$ in eq. (4.4) is replaced by the pseudo inverse of $(\mathbf{S}\mathbf{A})^T$, then the following biased estimate is obtained

$$\lambda \left\| \boldsymbol{\Sigma}_s^{-1}\mathbf{V}_s^T\mathbf{x}(\lambda) \right\|_2 = \left\| (\mathbf{S}\mathbf{A})^{\ddagger}\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)) \right\|_2 = \left\| \boldsymbol{\Sigma}_1^{-1}\mathbf{V}_1^T\mathbf{U}^T(\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)) \right\|_2, \tag{4.5}$$

where $\mathbf{U}_s\boldsymbol{\Sigma}_s\mathbf{V}_s^T$ and $\mathbf{U}_1\boldsymbol{\Sigma}_1\mathbf{V}_1^T$ denote the SVD of $\mathbf{S}\mathbf{A} \in \mathbb{R}^{m\times d}$ and $\mathbf{S}\mathbf{U}$ matrices, respectively. The bias of the estimate in eq. (4.5) is given by (Lemma 1 of [90]):

$$\mathbb{E}_{\mathbf{S}}\left[\left\| (\mathbf{S}\mathbf{A})^{\ddagger}\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)) \right\|_2 \right] = \theta \left\| \mathbf{U}^T(\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)) \right\|_2,$$

where θ is a positive real number that does not dependent on λ, hence the approximated residual in eq. (4.5) maintains the same behaviour as the projected residual $\left\| \mathbf{U}^T(\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)) \right\|_2$ for varying λ values in the expectation. As for a single realization of the sketching matrix $\mathbf{S}$, the approximated residual in eq. (4.5) is still expected to behave similar to $\left\| \mathbf{U}^T(\mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)) \right\|_2$ for the varying λ because the multiplication with $\boldsymbol{\Sigma}_1^{-1}\mathbf{V}_1^T$ corresponds to a random scaling of the residual vector in random directions. If the scaling was on directions of the left singular vectors of $\mathbf{A}$, i.e., if the random rotation $\mathbf{V}_1^T$ in eq. (4.5) was not present, then the multiplication with $\boldsymbol{\Sigma}_1^{-1}$ would substantially distort the parameter selection mechanism explained in Section 2.2.2 since the energy of the spectral terms would be significantly altered by $\boldsymbol{\Sigma}_1^{-1}$. Such deformations are prevented in eq. (4.5) since, by pre-multiplication with $\mathbf{V}_1^T$, the impact of scaling is dispersed over the spectral terms and becomes negligible. To show the effect of the random scaling and the rotation, we conducted the following numerical experiment: we generated a seismic travel-time tomography problem via IRtool [11]. The image size was set to 16×16, number of sources and receivers were set to 64, which gives an over-determined coefficient matrix $\mathbf{A} \in \mathbb{R}^{4096\times 256}$. The noise level was set to $\frac{\|\mathbf{w}\|_2}{\|\mathbf{A}\mathbf{x}_0\|_2} = 5\%$. Then, 1000 samples of Gaussian sketches $\mathbf{S}_i \in \mathbb{R}^{768\times 4096}$ were created and the SVD $\mathbf{S}_i\mathbf{U} = \mathbf{U}_i\boldsymbol{\Sigma}_i\mathbf{V}_i^T$ was computed for each sample. The corresponding GCV curves, i.e., $\frac{\left\|\boldsymbol{\Sigma}_i^{-1}\mathbf{V}_i^T\mathbf{U}^T(\mathbf{b}-\mathbf{A}\mathbf{x}(\lambda))\right\|_2^2}{\mathsf{tr}(\mathbf{I}_d - P_{\boldsymbol{\Sigma}}(\lambda))^2}$ and $G_{full}(\lambda, d) = \frac{\left\|\mathbf{U}^T(\mathbf{b}-\mathbf{A}\mathbf{x}(\lambda))\right\|_2^2}{\mathsf{tr}(\mathbf{I}_d - P_{\boldsymbol{\Sigma}}(\lambda))^2}$ were normalized to the interval [0, 1]. As shown in Figure 4.2, the GCV curves of a total of 1000 approximations and the $G_{full}(\lambda, d)$ are almost identical. The subplot under the main plot shows the histogram of the minimizer of the approximated GCV functions which are densely concentrated around the λ_d^{gcv} value. The insert in Figure 4.2 shows the histogram of the relative oracle error of $\mathbf{x}(\lambda_i)$ where λ_i is the minimizer of the approximated GCV function that is constructed by $\mathbf{S}_i$. The small difference between λ_i's and λ_d^{gcv} does not cause a significant deviation in $\mathbf{x}(\lambda_i)$, moreover 600 minimizers out of 1000 approximated GCV functions produce even better results than the one produced by the λ_d^{gcv}. The error produced by approximately 200 of the remaining minimizers are only larger than the error produced by λ_d^{gcv} at the third decimal of the logarithmic base. The overall graph shows that the random scaling and the rotation stemming from the multiplication by $\boldsymbol{\Sigma}_1^{-1}\mathbf{V}_1^T$ do not have a significant impact on the parameter selection mechanism, therefore it can be neglected. The caveat here is that although the sketch size m can be chosen proportional to $\mathsf{sd}_\lambda(\mathbf{A})$ for the naive `M-IHS` technique, the entries of the diagonal matrix $\boldsymbol{\Sigma}_1^{-1}$ can be kept in the interval $[1/(1+\epsilon),\ 1/(1-\epsilon)]$ for a plausible $\epsilon \in (0,\ 1)$ value with high probability as long as the sketch size m for the `Hybrid M-IHS` is chosen larger than d [53, 69].

To find a proper regularization parameter for the HS step at the i^{th} iteration, we substitute $\mathbf{x}^i + \Delta\mathbf{x}^i(\lambda)$ for $\mathbf{x}(\lambda)$ in eq. (4.5), in that way we avoid access to data pair $\mathbf{A}, \mathbf{b}$ for the estimation purpose. Moreover, we use the singular values $\boldsymbol{\Sigma}_s$ of the sketched matrix for the trace term $\mathsf{tr}\left(P_{\boldsymbol{\Sigma}}(\lambda)\right)$ in the denominator. Unlike the singular values of the bidiagonal form $\mathbf{B}_k$, the singular values of the sketched matrix $\mathbf{SA}$ are accurate estimates for the singular values of $\mathbf{A}$. The regularization parameter λ_i for the i^{th} update in eq. (4.3) is chosen as the minimizer of the

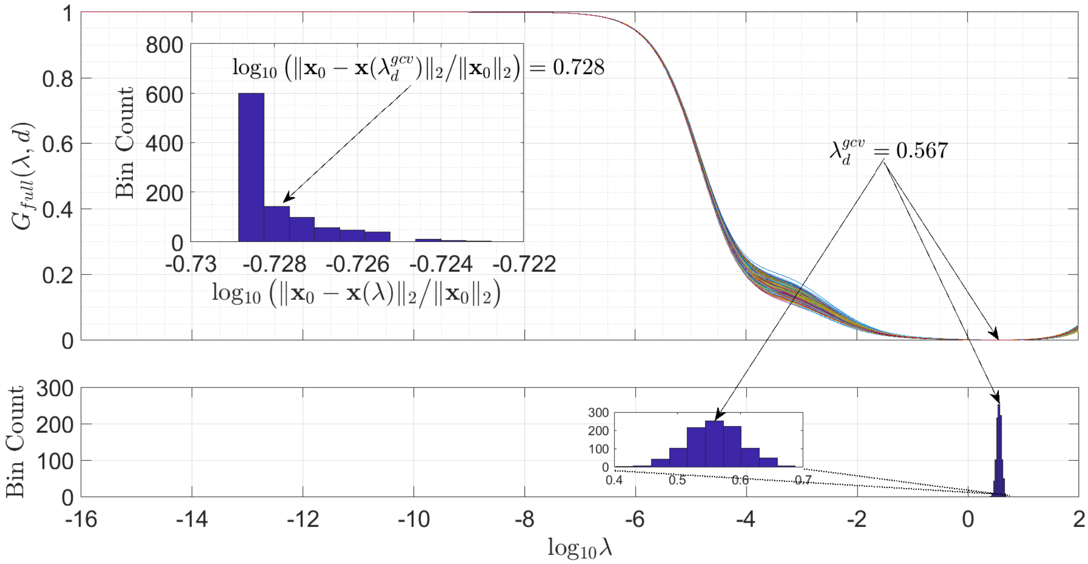


Figure 4.2: *The impact of the random scaling and the rotation on the parameter selection.*

following risk function:

$$\mathbb{V}_1(\lambda) = \frac{\lambda \left\|\mathbf{\Sigma}_s^{-1}\mathbf{V}_s^T\left(\mathbf{x}^i + \Delta\mathbf{x}^i(\lambda)\right)\right\|_2}{d - \mathsf{tr}\left(P_{\mathbf{\Sigma}_s}(\lambda)\right)}, \tag{4.6}$$

which can also be derived by using a linear system interpretation as follows. The un-regularized version of the linear system in eq. (4.3), i.e., the linear system that we aim to regularize, is

$$(\mathbf{SA})^T(\mathbf{SA})\Delta\mathbf{x}^i = \mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i) := \mathbf{g}^i. \tag{4.7}$$

As opposed to the projected problem obtained by the conventional hybrid methods, the GCV (or any other predictive risk based estimator) cannot be directly applied to the linear system in eq. (4.7) because of multiplication with $\mathbf{A}^T$ that scales the spectral terms by its singular values. To alleviate the effect of scaling with the singular values of $\mathbf{A}^T$ on the RHS of eq. (4.7) and to obtain a reliable estimate of the residual error at an affordable complexity, instead of the pseudo-inverse of $\mathbf{A}^T$, the pseudo inverse of $(\mathbf{SA})^T$ can be used as:

$$\mathbf{SA}\Delta\mathbf{x}^i = (\mathbf{SA})^{\ddagger}\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i) = \mathbf{U}_1\mathbf{\Sigma}_1^{-1}\mathbf{V}_1^T\mathbf{U}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i), \tag{4.8}$$

which is equivalent to

$$\mathbf{\Sigma}_s \mathbf{V}_s^T \Delta \mathbf{x}^i = \mathbf{\Sigma}_s^{-1} \mathbf{V}_s^T \mathbf{g}^i. \tag{4.9}$$

If the effect of multiplication with $\mathbf{\Sigma}_1^{-1}\mathbf{V}_1^T$ on the residual vector is neglected as discussed earlier, then we get a linear transformation between the measurements (or the residual) and the current HS step $\Delta\mathbf{x}^i$. Thus, we can apply the GCV on the system given in eq. (4.9) as

$$\mathbb{V}_1(\lambda) = \frac{\left\| \mathbf{\Sigma}_s^{-1} \mathbf{V}_s^T \mathbf{g}^i - \mathbf{\Sigma}_s \mathbf{V}_s^T \Delta \mathbf{x}^i(\lambda) \right\|_2}{d - \mathsf{tr}\left(P_{\mathbf{\Sigma}_s}(\lambda)\right)}, \tag{4.10}$$

where $\Delta\mathbf{x}^i(\lambda)$ is the regularized solution of the linear system given in eq. (4.7):

$$\Delta \mathbf{x}^i(\lambda) = \left((\mathbf{SA})^T (\mathbf{SA}) + \lambda \mathbf{I}_d \right)^{-1} \left(\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}^i) - \lambda \mathbf{x}^i \right).$$

Note that, the risk functions given in eq. (4.6) and eq. (4.10) are equivalent to each other and both are used for the derivation purposes only. In Algorithm 8, we give an efficient form that requires fewer operations than those given in eq. (4.6) and eq. (4.10).

After obtaining a proper estimate for the λ_i by minimizing the risk function $\mathbb{V}_1(\lambda)$, the momentum parameters α_i and β_i can be selected as

$$\beta_i = \mathsf{sd}_{\lambda_i}(\mathbf{\Sigma}_s)/m, \qquad \alpha_i = (1 - \beta_i)^2, \tag{4.11}$$

which empirically maximize the convergence rate of the i^{th} iteration to the solution $\mathbf{x}(\lambda_i)$ (Corollary 3.4 of [37]). When different regularization and momentum parameters are used for each iteration, the convergence of the `Hybrid M-IHS` updates given in eq. (4.3) to the solution is characterized by the following theorem.

Theorem 4.2.1. *Let $\mathbf{x}(\mathbf{\Phi})$ is the regularized solution with filtering coefficients $\mathbf{\Phi} = \mathbf{diag}(\phi_1, \ldots, \phi_r)$ as defined in eq. (2.4). Assume $\mathcal{T}_i$ is an update rule depending on the filtering coefficients $\mathbf{\Phi}_i$ and the momentum parameters α_i, β_i such that*

$$\left\| \mathcal{T}_i(\mathbf{x}^i) - \mathbf{x}(\mathbf{\Phi}_i) \right\|_2 \leq \rho_i \left\| \mathbf{x}^i - \mathbf{x}(\mathbf{\Phi}_i) \right\|_2, \tag{4.12}$$

where $\mathbf{x}^{i+1} = \mathcal{T}_i(\mathbf{x}^i) = \mathcal{T}_i(\mathcal{T}_{i-1}(\ldots \mathcal{T}_0(\mathbf{x}^0)))$. *Then, the following error bound holds:*

$$\left\|\mathbf{x}^{i+1} - \mathbf{x}(\mathbf{\Phi}_i)\right\|_2 \leq \left(\prod_{j=0}^{i} \rho_j\right) \left\|\mathbf{x}^0 - \mathbf{x}(\mathbf{\Phi}_0)\right\|_2 + \sum_{j=1}^{i} \left(\prod_{\ell=j}^{i} \rho_\ell\right) \left\|\mathbf{x}(\mathbf{\Phi}_j) - \mathbf{x}(\mathbf{\Phi}_{j-1})\right\|_2 .$$

Proof. The result of the theorem is obtained by using the condition in eq. (4.12) and application of the triangle inequality as:

$$\begin{aligned}
\left\|\mathbf{x}^{i+1} - \mathbf{x}(\mathbf{\Phi}_i)\right\|_2 &\leq \rho_i \left\|\mathbf{x}^i - \mathbf{x}(\mathbf{\Phi}_i)\right\|_2 = \rho_i \left\|\mathbf{x}^i - \mathbf{x}(\mathbf{\Phi}_{i-1}) + \mathbf{x}(\mathbf{\Phi}_{i-1}) - \mathbf{x}(\mathbf{\Phi}_i)\right\|_2 \\
&\leq \rho_i \left\|\mathbf{x}^i - \mathbf{x}(\mathbf{\Phi}_{i-1})\right\|_2 + \rho_i \left\|\mathbf{x}(\mathbf{\Phi}_{i-1}) - \mathbf{x}(\mathbf{\Phi}_i)\right\|_2 \\
&\leq \rho_i \rho_{i-1} \left\|\mathbf{x}^{i-1} - \mathbf{x}(\mathbf{\Phi}_{i-1})\right\|_2 + \rho_i \left\|\mathbf{x}(\mathbf{\Phi}_{i-1}) - \mathbf{x}(\mathbf{\Phi}_i)\right\|_2 \\
&\leq \rho_i \rho_{i-1} \left\|\mathbf{x}^{i-1} - \mathbf{x}(\mathbf{\Phi}_{i-2})\right\|_2 + \rho_i \rho_{i-1} \left\|\mathbf{x}(\mathbf{\Phi}_{i-2}) - \mathbf{x}(\mathbf{\Phi}_{i-1})\right\|_2 \\
&\qquad + \rho_i \left\|\mathbf{x}(\mathbf{\Phi}_{i-1}) - \mathbf{x}(\mathbf{\Phi}_i)\right\|_2 .
\end{aligned}$$

Successively, repeating this process for i times produces the desired result. □

If the update rule proposed in eq. (4.3) is used, then the assumption in eq. (4.12) is satisfied by Theorem 3.1 of [37] with a $\rho_i < 1$. The convergence rate ρ_i is empirically modeled by the ratio $\mathtt{sd}_{\lambda_i}(\mathbf{A})/m$ [37]. For simplicity assume that the parameter sequence $\{\lambda_j\}_{j=1}^{i}$ that is selected through the proposed risk function satisfies $\lambda_0 \geq \ldots \geq \lambda_i$, and the initial estimate $\mathbf{x}^0$ is chosen to be zero, then the error upper bound in Theorem 4.2.1 reduces to

$$\left\|\mathbf{x}^{i+1} - \mathbf{x}(\lambda_i)\right\|_2 \leq \sum_{j=0}^{i} (\rho_i)^{i+1-j} \left\|\mathbf{x}(\lambda_j) - \mathbf{x}(\lambda_{j-1})\right\|_2 , \tag{4.13}$$

where $\mathbf{x}(\lambda_{-1}) = \mathbf{0}$ and $\rho_i \geq \rho_j$ for $\lambda_i \leq \lambda_j$. Each term in the summation eq. (4.13) corresponds to the norm of the filtered spectral terms that differ from iteration to iteration and the value of the summation can be reduced arbitrarily by using the same λ_i for a few additional iterations. This upper bound means that recovery of the true input components $\mathbf{V}_{k'}\mathbf{V}_{k'}^T\mathbf{x}_0$ for some $1 \leq k' \leq r$ continue during the iterations that the regularization parameters do not filter the corresponding singular vectors. Hence, as long as the regularization parameters used in eq. (4.3) are greater than the λ^{gcv}, each iteration of the `Hybrid M-IHS` increases the accuracy of some parts in the solution estimate. This working dynamic becomes clearer when the tail of the Tikhonov coefficients is ignored and a hard-thresholding scheme that is similar to the TSVD is used in the regularization. In such cases, the error bound in eq. (4.13) reduces to

$$\left\|\mathbf{x}^{i+1} - \mathbf{x}(k_i)\right\|_2 \leq \sum_{j=0}^{i} \left(\sqrt{\frac{k_i}{m}}\right)^{i+1-j} \left\|\mathbf{x}(k_j) - \mathbf{x}(k_{j-1})\right\|_2 , \tag{4.14}$$

where $k_{-1} = 0 \leq k_1 \leq \ldots \leq k_i$ and the truncation parameter k_j with momentum parameters $\beta_j = \frac{k_j}{m}$ and $\alpha_j = (1-\beta_j)^2$ are used in the j^{th} iteration. An algorithm that realizes the above bound is proposed in Appendix B, but it is not efficient enough for practical use.

Algorithm 8 `Hybrid M-IHS` (for $n \gg d$)

```
Input: $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{b}$, $m$, $\mathbf{x}^0$                         complexity
    $\mathbf{SA} = \texttt{RP\_fun}(\mathbf{A}, m)$                                        $C(n,d,m)$
[$\mathbf{\Sigma}_s, \mathbf{V}_s$] = svd($\mathbf{SA}$)                                          $O(md^2)$
while until stopping criteria do
    $\mathbf{g}^i = \mathbf{V}_s^T \mathbf{A}^T (\mathbf{b} - \mathbf{A}\mathbf{x}^i)$                                         $O(nd)$
    $\mathbf{f}^i = \mathbf{\Sigma}_s^{-1}\mathbf{g}^i + \mathbf{\Sigma}_s \mathbf{V}_s^T \mathbf{x}^i$                                        $O(d^2)$
    $\lambda_i = \underset{\lambda}{\operatorname{argmin}} \left\|(\mathbf{\Sigma}_s^2 + \lambda\mathbf{I})^{-1}\mathbf{f}^i\right\|_2 / \mathsf{tr}((\mathbf{\Sigma}_s^2 + \lambda\mathbf{I})^{-1})$   $O(d)$
  $\Delta\mathbf{x}^i = \mathbf{V}_s (\mathbf{\Sigma}_s^2 + \lambda_i\mathbf{I})^{-1} (\mathbf{g}^i - \lambda_i \mathbf{V}_s^T \mathbf{x}^i)$                     $O(d^2)$
    $\widehat{k} = d - \lambda_i \mathsf{tr}\left((\mathbf{\Sigma}_s^2 + \lambda_i\mathbf{I})^{-1}\right)$                          $O(d)$
    $\beta_i = \widehat{k}/m$
    $\alpha_i = (1-\beta_i)^2$
  $\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha_i\Delta\mathbf{x}^i + \beta_i(\mathbf{x}^i - \mathbf{x}^{i-1})$                           $O(d)$
end while
```

For dense coefficient matrices, the overall computational complexity of Algorithm 8 is $O(nd\log(m) + Nnd + md^2)$ where N is the number of iterations, m is the sketch size that is proportional to d and the ROS sketch matrices are assumed to be used (for other choices see [55, 37]). The $O(md^2)$ term, which is due to the SVD, cannot be avoided but the required computation in practice can be significantly reduced by using the GKL procedure as given in Algorithm 12. The main computational advantage of the `Hybrid M-IHS` over the conventional techniques is the saving in the number of gradient computations N that is almost independent of the spectral properties as opposed to the effective rank k^*. In distributed memory environments, the `Hybrid M-IHS` requires only one distributed multiplication per iteration due to the gradient computation and it computes the decomposition (SVD or GKL) of an $m \times d$ dimensional sketched matrix $\mathbf{SA}$ only once prior to the start of iterations in a single node of the memory network. Thus, parallel implementation of the preferred decomposition that, for example, run on the GPUs [91] can be readily adapted into the algorithm. The memory space required by Algorithm 8 scales as $O(d^2)$. If $\mathbf{A}$ is a sparse matrix, then significant savings can be gained by careful implementation. For sparse cases, complexity of the gradient computation $O(Nnd)$ reduces to $O(N\text{nnz}(\mathbf{A}))$. For such cases data oblivious sparse sketching matrices such as CountSketch with run-time of $O(\text{nnz}(\mathbf{A}))$ can be preferred and the complexity of the bidiagonalization procedure together with the re-orthogonalization steps becomes $O(d^3 + \text{nnz}(\mathbf{A})d)$, so the overall complexity is reduced down to $O((N+d)\text{nnz}(\mathbf{A}) + d^3)$. If $\mathbf{A}$ is an operator, without any modification the `Hybrid M-IHS` can still be used.

4.2.2 Hybrid Dual M-IHS for highly under-determined problems

If the system in eq. (2.1) is under-determined, instead of the regularized objective function in eq. (2.3), the naive `Dual M-IHS` solves the dual problem given in eq. (3.12) and recovers the solution of the primal problem via the relation in eq. (3.13). Here, the proposed `Hybrid Dual M-IHS` uses the following update to solve the dual problem:

$$\Delta\boldsymbol{\nu}^i(\lambda_i) = \underset{\boldsymbol{\nu}\in\mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\left\|\mathbf{SA}^T\boldsymbol{\nu}\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{\nu}\right\|_2^2 + \langle\nabla g(\boldsymbol{\nu}^i, \lambda_i),\ \boldsymbol{\nu}\rangle \tag{4.15}$$

$$\boldsymbol{\nu}^{i+1} = \boldsymbol{\nu}^i + \alpha_i\Delta\boldsymbol{\nu}^i(\lambda_i) + \beta_i(\boldsymbol{\nu}^i - \boldsymbol{\nu}^{i-1}). \tag{4.16}$$

The difference from the naive `Dual M-IHS` technique is due to the varying regularization and momentum parameters. For the naive `Dual M-IHS`, the regularization parameter should be known prior to the iterations but here we obtain a proper estimate of the regularization parameter for the updates given in eq. (4.15) and adjust the momentum parameters accordingly as in the case of `Hybrid M-IHS`. For this purpose, we will use the fact that the dual solution scaled with the regularization parameter corresponds the residual error of the primal problem, i.e., $\lambda\boldsymbol{\nu}(\lambda) = \mathbf{b} - \mathbf{A}\mathbf{x}(\lambda)$, which enables us to write the GCV risk function given in eq. (2.8) as

$$G_{full}(\lambda) = \frac{\lambda \left\|\boldsymbol{\nu}(\lambda)\right\|_2}{\mathsf{tr}\left(\mathbf{I}_n - P_{\boldsymbol{\Sigma}}(\lambda)\right)}. \tag{4.17}$$

To find a proper regularization parameter for the *dual* HS step at the i^{th} iteration given in eq. (4.15), we substitute $\boldsymbol{\nu}^i + \Delta\boldsymbol{\nu}^i(\lambda)$ for $\boldsymbol{\nu}(\lambda)$ in eq. (4.17) and similar to over-determined case, we use the singular values of the sketched matrix in the denominator to estimate the degrees of freedom. Consequently, the regularization parameter λ_i is chosen as the minimizer of the following risk function

$$\mathbb{V}_2(\lambda) = \frac{\lambda \left\|\boldsymbol{\nu}^i + \Delta\boldsymbol{\nu}^i(\lambda)\right\|_2}{\mathsf{tr}\left(\mathbf{I}_n - P_{\boldsymbol{\Sigma}_s}(\lambda)\right)}, \tag{4.18}$$

which can be also derived from a different perspective by using a linear system interpretation. The un-regularized linear system in eq. (4.15), i.e., the linear system that we aim to regularize, is

$$(\mathbf{S}\mathbf{A}^T)^T(\mathbf{S}\mathbf{A}^T)\Delta\boldsymbol{\nu}^i = \mathbf{b} - \mathbf{A}\mathbf{A}^T\boldsymbol{\nu}^i := \mathbf{h}^i. \tag{4.19}$$

The RHS of eq. (4.19) is equal to the residual error of the primal problem, i.e., $\mathbf{h}^i = \mathbf{b} - \mathbf{A}\mathbf{x}^i$. As opposed to the `Hybrid M-IHS`, we have a linear transformation between the measurements and the dual HS step without requiring a matrix inversion. Therefore, the GCV can be applied directly to the linear system in eq. (4.19) as

$$\mathbb{V}_2(\lambda) = \frac{\left\|\mathbf{h}^i - \mathbf{V}_s\boldsymbol{\Sigma}_s^2\mathbf{V}_s^T\Delta\boldsymbol{\nu}^i(\lambda)\right\|_2}{\mathsf{tr}\left(\mathbf{I}_n - P_{\boldsymbol{\Sigma}_s}(\lambda)\right)}, \tag{4.20}$$

where $\mathbf{U}_s\boldsymbol{\Sigma}_s\mathbf{V}_s^T$ denotes the SVD of the sketched matrix $\mathbf{A}\mathbf{S}^T \in \mathbb{R}^{n\times m}$ and $\boldsymbol{\nu}(\lambda)$ is the regularized solution of the linear system given in eq. (4.19):

$$\Delta\boldsymbol{\nu}^i(\lambda) = \left((\mathbf{S}\mathbf{A}^T)^T(\mathbf{S}\mathbf{A}^T) + \lambda\mathbf{I}\right)^{-1}\left(\mathbf{b} - \mathbf{A}\mathbf{A}^T\boldsymbol{\nu}^i - \lambda\boldsymbol{\nu}^i\right).$$

Both functions in eq. (4.18) and eq. (4.20) are equivalent. In Algorithm 9, we give an efficient form of the risk function that requires less operations than those given in eq. (4.18) and eq. (4.20). When the parameter λ_i is selected by minimizing $\mathbb{V}_2(\lambda)$, the momentum parameters are chosen in the same fashion as given in eq. (4.11). As in the convergence analysis of the `Hybrid M-IHS`, convergence properties of the `Hybrid Dual M-IHS` can straightforwardly be characterized by using Theorem 3.1 of [37].

Algorithm 9 `Hybrid Dual M-IHS` (for $n \ll d$)

```
 1: Input: $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{b}$, $m$                                complexity
 2:     $\mathbf{SA}^T = \texttt{RP\_fun}(\mathbf{A}^T, m)$
 3: $[\mathbf{\Sigma}_s,\ \mathbf{V}_s] = \texttt{svd}(\mathbf{SA}^T,\ n)$                          $O(mn^2)$
 4:      $\boldsymbol{\nu}^0 = \mathbf{x}^0 = \mathbf{0}$
 5: while until stopping criteria do
 6:     $\widetilde{\mathbf{h}}^i = \mathbf{V}_s^T \left(\mathbf{b} - \mathbf{A}\mathbf{x}^i\right)$                          $O(nd)$
 7:     $\mathbf{f}^i = \widetilde{\mathbf{h}}^i + \mathbf{\Sigma}_s^2 \mathbf{V}_s^T \boldsymbol{\nu}^i$                             $O(n^2)$
 8:     $\lambda_i = \underset{\lambda}{\operatorname{argmin}} \left\| \left(\mathbf{\Sigma}_s^2 + \lambda \mathbf{I}\right)^{-1} \mathbf{f}^i \right\|_2 / \mathsf{tr}\left((\mathbf{\Sigma}_s^2 + \lambda \mathbf{I})^{-1}\right)$        $O(n)$
 9:   $\Delta\boldsymbol{\nu}^i = \mathbf{V}_s \left(\mathbf{\Sigma}_s^2 + \lambda_i \mathbf{I}_d\right)^{-1} \left(\widetilde{\mathbf{h}}^i - \lambda_i \mathbf{V}_s^T \boldsymbol{\nu}^i\right)$          $O(n^2)$
10:     $\widehat{k} = d - \lambda_i \mathsf{tr}\left(\left(\mathbf{\Sigma}_s^2 + \lambda_i \mathbf{I}\right)^{-1}\right)$                       $O(n)$
11:     $\beta_i = \widehat{k}/m$
12:     $\alpha_i = (1 - \beta_i)^2$
13:   $\boldsymbol{\nu}^{i+1} = \boldsymbol{\nu}^i + \alpha_i \Delta\boldsymbol{\nu}^i + \beta_i(\boldsymbol{\nu}^i - \boldsymbol{\nu}^{i-1})$                        $O(n)$
14:   $\mathbf{x}^{i+1} = \mathbf{A}^T \boldsymbol{\nu}^{i+1}$                                              $O(nd)$
15: end while
```

For the problems with dense coefficient matrices, the computational complexity of the `Hybrid Dual M-IHS` is $O(nd\log(m) + Nnd + mn^2)$, where m is proportional to $\mathsf{sd}_\lambda(\mathbf{A})$ and the ROS sketching matrices is assumed. The GKL bidiagonalization can be used to further reduce the complexity $O(mn^2)$ of the SVD as shown in Algorithm 13. *Line 14* of Algorithm 9 can be incorporated into *Line 6* so that only one distributed matrix-vector computation with coefficient matrix $\mathbf{A}$ is required per iteration. The `Hybrid Dual M-IHS` maintains the same computational advantages that the `M-IHS` has over the conventional hybrid methods. In sparse problems the complexity of Algorithm 9 can be reduced to $O((N+n)\text{nnz}(\mathbf{A}) + n^3)$

in the same way as the highly over-determined case. The total memory space used by Algorithm 9 scales as $O(n^2)$.

4.2.3 Hybrid Primal Dual M-IHS for square-like problems

Assume that the effective rank k^* of the problem is substantially smaller than both size of the coefficient matrix, i.e., $k^* \ll n, d$, where $n > d$, $n = d$ or $n < d$. From the discussion about Tikhonov and the TSVD regularization, we know that $k^* \approx \mathsf{sd}_\lambda(\mathbf{A})$ for a properly chosen regularization parameter λ and as discussed in Section 2.2.2.1, a proper estimate of λ can be obtained by using $G_{full}(\lambda, k)$ function given in eq. (2.9) if k is sufficiently larger than k^*. Assume also we have a rough estimate k' such that $k' > k^*$ so that the sketch size for the `Hybrid Dual M-IHS` can be chosen proportional to the statistical dimension, e.g., $m_1 = 2k'$. However, $O(nm_1^2)$ complexity of $n \times m_1$ dimensional matrix decomposition that is used for the minimization of $\mathbb{V}_2(\lambda)$ function might still be computationally prohibitive since both n and d might be very large and scale similar. In such cases, the dual of the sub-problem solved in eq. (4.15) corresponds to solving a highly over-determined linear system consists of the data pair $(\mathbf{AS}^T, \nabla g(\boldsymbol{\nu}^i, \lambda))$ since $k' \ll n, d$. To see this, consider the following LS problem that is solved for the dual variable $\mathbf{z}^i(\lambda)$ of the dual HS step $\Delta\boldsymbol{\nu}^i(\lambda)$ given in eq. (4.15)

$$\mathbf{z}^i(\lambda) = \underset{\mathbf{z}\in\mathbb{R}^{m_1}}{\operatorname{argmin}} \; \underbrace{\left\|\mathbf{AS}^T\mathbf{z} + \nabla g(\boldsymbol{\nu}^i, \lambda)\right\|_2^2 + \lambda \left\|\mathbf{z}\right\|_2^2}_{h(\mathbf{z}, \boldsymbol{\nu}^i, \lambda)}, \tag{4.21}$$

where the relation between the two variables is

$$\Delta\boldsymbol{\nu}^i(\lambda) = -\left(\nabla g(\boldsymbol{\nu}^i, \lambda) + \mathbf{AS}^T\mathbf{z}^i(\lambda)\right)/\lambda \iff \mathbf{z}^i(\lambda) = \mathbf{SA}^T\Delta\boldsymbol{\nu}^i(\lambda). \tag{4.22}$$

Therefore, the parameter λ_i and the solution $\boldsymbol{\nu}(\lambda_i)$ at the i^{th} iteration of the `Hybrid Dual M-IHS` can be iteratively obtained by applying another dimension reduction on the problem in eq. (4.21) via the `Hybrid M-IHS` technique as follows

$$\begin{aligned}\Delta\mathbf{z}^{i,j}(\lambda_{i,j}) &= \underset{\mathbf{z}\in\mathbb{R}^{m_1}}{\operatorname{argmin}} \;\; \left\|\mathbf{WAS}^T\mathbf{z}\right\|_2^2 + \lambda_{i,j}\left\|\mathbf{z}\right\|_2^2 + 2\langle\nabla_{\mathbf{z}}h(\mathbf{z}^{i,j},\boldsymbol{\nu}^i,\lambda_{i,j}),\ \mathbf{z}\rangle, \quad (4.23)\\ \mathbf{z}^{i,j+1} &= \mathbf{z}^{i,j} + \alpha_j\Delta\mathbf{z}^{i,j}(\lambda_{i,j}) + \beta_j(\mathbf{z}^{i,j} - \mathbf{z}^{i,j-1}),\end{aligned}$$

where j is the index of the inner iterations conducted at each update with index i given in eq. (4.15), $\lambda_{i,j}$ is the estimate of λ_i at the j^{th} inner iteration and $\mathbf{W}\in\mathbb{R}^{m_2\times n}$ is the second sketching matrix. The inexact dual HS step $\Delta\boldsymbol{\nu}^i$ can be computed by substituting $\mathbf{z}^{i,M}$ for $\mathbf{z}^i(\lambda)$ in eq. (4.22) where M is the number of inner iterations. In that way, instead of $n\times m_1$ dimensional matrix decomposition with a computational compelxity of $O(nm_1^2)$, an $m_2\times m_1$ dimensional matrix decomposition can be used with a significantly lower computational compelxity of $O((k')^3)$, where m_2 is a few times larger than m_1 as discussed in Section 4.2.1, e.g., $m_2 = 2m_1 = 4k'$.

The difference of the update given in eq. (4.23) from the naive `Primal Dual M-IHS` techniques proposed in [37] is the varying regularization and momentum parameters. The regularization parameter should be known prior to the iterations of the naive `Primal Dual M-IHS` techniques, but here we aim to obtain a proper regularization parameter for the inner most HS step given in eq. (4.23) and to tune the momentum parameters accordingly. To obtain an estimate for the regularization parameter, we will combine the biased estimate idea used in eq. (4.5) with the risk function $\mathbb{V}_2(\lambda)$ given in eq. (4.18).

In the `Hybrid Primal Dual M-IHS` iterations, the dual solution $\boldsymbol{\nu}(\lambda)$ is estimated over two nested loops as $\boldsymbol{\nu}^i - \nabla g(\boldsymbol{\nu}^i,\lambda) - \mathbf{AS}^T(\mathbf{z}^{i,j} + \Delta\mathbf{z}^{i,j}(\lambda))$. Hence, through the same approach as the `Hybrid Dual M-IHS`, a proper estimate $\lambda_{i,j}$ for the regularization parameter can be obtained by substituting the solution estimate of $\boldsymbol{\nu}(\lambda)$ at the j^{th} inner iteration of the i^{th} outer loop for $\boldsymbol{\nu}(\lambda)$ in the GCV function given in eq. (4.17) as

$$\mathbb{V}_2(\lambda) = \frac{\lambda\left\|\boldsymbol{\nu}^i - \nabla g(\boldsymbol{\nu}^i,\lambda) - \mathbf{AS}^T(\mathbf{z}^{i,j} + \Delta\mathbf{z}^{i,j}(\lambda))\right\|_2}{\mathsf{tr}\left(\mathbf{I}_n - P_{\boldsymbol{\Sigma}_s}(\lambda)\right)}, \qquad (4.24)$$

where $\mathbf{U}_s\mathbf{\Sigma}_s\mathbf{V}_s^T$ is the SVD of $\mathbf{AS}^T$. This requires to access $n \times m_1$ dimensional $\mathbf{AS}^T$ matrix for each λ value used in the minimization of $\mathbb{V}_2(\lambda)$, which is undesirable for large n. Therefore instead of the risk function in eq. (4.24), similar to the approach used in eq. (4.5) to derive $\mathbb{V}_1(\lambda)$, we will use the following biased estimate of the residual error that is projected onto the range space of $\mathbf{AS}^T$ which is a close approximation to the dominant k' dimensional range space of $\mathbf{A}$ (see Proto-Algorithm and its analysis given in [92])

$$\lambda \left\|\mathbf{\Sigma}_w^{-1}\mathbf{V}_w^T\mathbf{SA}^T\boldsymbol{\nu}(\lambda)\right\|_2 = \left\|\mathbf{\Sigma}_w^{-1}\mathbf{V}_w^T\mathbf{SA}^T(\mathbf{b}-\mathbf{Ax}(\lambda))\right\|_2 = \left\|\mathbf{\Sigma}_2^{-1}\mathbf{V}_2^T\mathbf{U}_s^T(\mathbf{b}-\mathbf{Ax}(\lambda))\right\|_2 \tag{4.25}$$

where $\mathbf{U}_w\mathbf{\Sigma}_w\mathbf{V}_w^T$ and $\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^T$ are the SVD of $\mathbf{WAS}^T$ and $\mathbf{WU}_s$, respectively. The bias of the estimate is given by

$$\mathbb{E}_{\mathbf{W}}\left[\left\|(\mathbf{WAS})^{\ddagger}\mathbf{SA}^T(\mathbf{b}-\mathbf{Ax}(\lambda))\right\|_2\right] = \theta \left\|\mathbf{U}_s^T(\mathbf{b}-\mathbf{Ax}(\lambda))\right\|_2, \tag{4.26}$$

where θ is a positive real number that does not dependent on λ. Therefore, the residual in eq. (4.26) maintains the same behaviour as the residual $\left\|\mathbf{U}_s^T(\mathbf{b}-\mathbf{Ax}(\lambda))\right\|_2$ for varying λ in the expectation. Multiplying the projected residual error with $\mathbf{\Sigma}_2^{-1}\mathbf{V}_2^T$ has the same negligible effect on the parameter selection mechanism as in the case of the `Hybrid M-IHS` which is demonstrated in Figure 4.2.

To find a proper $\lambda_{i,j}$ estimate we first substitute outer loop solution estimate $\boldsymbol{\nu}^i + \Delta\boldsymbol{\nu}^i(\lambda)$ for $\boldsymbol{\nu}(\lambda)$ in eq. (4.25) and then $\mathbf{z}^{i,j} + \Delta\mathbf{z}^{i,j}(\lambda)$ for $\mathbf{SA}^T\Delta\boldsymbol{\nu}^i(\lambda)$ due to the relation in eq. (4.22). As for the denominator that estimates the degrees of freedom in the residual error, we use singular values of $\mathbf{WAS}^T$ which gives the following risk estimator:

$$\mathbb{V}_3(\lambda) = \frac{\lambda\left\|\mathbf{\Sigma}_w^{-1}\mathbf{V}_w^T(\mathbf{SA}^T\boldsymbol{\nu}^i + \mathbf{z}^{i,j} + \Delta\mathbf{z}^{i,j}(\lambda))\right\|_2}{m_1 - \mathsf{tr}\left(P_{\mathbf{\Sigma}_w}(\lambda)\right)}. \tag{4.27}$$

Note that the last a few singular values of $\mathbf{\Sigma}_w = \mathbf{diag}(\sigma_{w,1}, \ldots, \sigma_{w,m_1})$ may underestimate the corresponding singular values of $\mathbf{A}$. Since m_1 is a few times larger than k^*, it does not cause an underestimation issue in the estimated degrees of freedom unlike the singular values of the bidiagonal matrix discussed in

Section 4.1. Also, the value m_1 takes the places of $\min(n, d)$ in the denominator, because the residual error is projected onto m_1 dimensional subspace. The risk function $\mathbb{V}_3(\lambda)$ can be derived by using a linear system interpretation as well. Consider the following un-regularized version of the linear system in eq. (4.23)

$$\left((\mathbf{WAS}^T)^T\mathbf{WAS}^T\right)\Delta\mathbf{z}^{i,j} = \mathbf{SA}^T\left(\mathbf{b} - \mathbf{Ax}^i - (\mathbf{SA}^T)^T\mathbf{z}^{i,j}\right) := \mathbf{g}^{i,j}, \tag{4.28}$$

where $\mathbf{h}^i$ is defined in eq. (4.19). The linear system in eq. (4.28) is in the same form as the linear system given in eq. (4.7) that is solved for the `Hybrid M-IHS`. Hence, similar to case of the `Hybrid M-IHS`, GCV-like risk functions cannot be directly applied to this system because multiplication of the RHS with $\mathbf{SA}^T$ scales the spectral terms. To alleviate the effect of the scaling and obtain a reliable estimate for the residual error with an affordable complexity, instead of the pseudo-inverse of $\mathbf{SA}^T$, the pseudo-inverse of $(\mathbf{WAS}^T)^T$ can be used as

$$\begin{aligned}\mathbf{WAS}^T\Delta\mathbf{z}^{i,j} &= (\mathbf{WAS}^T)^{\ddagger}\mathbf{SA}^T\left(\mathbf{b} - \mathbf{Ax}^i - (\mathbf{SA}^T)^T\mathbf{z}^{i,j}\right)\\ &= \mathbf{U}_2\boldsymbol{\Sigma}_2^{-1}\mathbf{V}_2^T\mathbf{U}_s^T\left(\mathbf{b} - \mathbf{Ax}^i - (\mathbf{SA}^T)^T\mathbf{z}^{i,j}\right),\end{aligned}$$

which is equivalent to

$$\boldsymbol{\Sigma}_w\mathbf{V}_w^T\Delta\mathbf{z}^{i,j} = \boldsymbol{\Sigma}_w^{-1}\mathbf{V}_w^T\mathbf{SA}^T\left(\mathbf{b} - \mathbf{Ax}^i - (\mathbf{SA}^T)^T\mathbf{z}^{i,j}\right) = \boldsymbol{\Sigma}_w^{-1}\mathbf{V}_w^T\mathbf{g}^{i,j}. \tag{4.29}$$

If the effect of the multiplication with $\boldsymbol{\Sigma}_2^{-1}\mathbf{V}_2^T$ is neglected in eq. (4.29) as discussed early, then we get a linear transformation between the measurements (or the residual) and the innermost HS step $\Delta\mathbf{z}^{i,j}$. Hence, we can apply the GCV on the linear system given in eq. (4.29) as

$$\mathbb{V}_3(\lambda) = \frac{\left\|\boldsymbol{\Sigma}_w^{-1}\mathbf{V}_w^T\mathbf{g}^{i,j} - \boldsymbol{\Sigma}_w\mathbf{V}_w^T\Delta\mathbf{z}^{i,j}(\lambda)\right\|_2}{\mathsf{tr}\left(\mathbf{I}_{m_1} - P_{\boldsymbol{\Sigma}_w}(\lambda)\right)}, \tag{4.30}$$

where $\Delta\mathbf{z}^{i,j}(\lambda)$ is the regularized solution of the linear system given in eq. (4.28):

$$\Delta\mathbf{z}^{i,j}(\lambda) = \left((\mathbf{WAS}^T)^T\mathbf{WAS}^T + \lambda\mathbf{I}_{m_1}\right)^{-1}\left(\mathbf{g}^{i,j} - \lambda(\mathbf{z}^{i,j} + \mathbf{SA}^T\boldsymbol{\nu}^i)\right).$$

Algorithm 10 `Hybrid Primal Dual M-IHS` (for $n \leq d$ or $n \geq d$)

```
Input: $\mathbf{A} \in \mathbb{R}^{n \times d},\ \mathbf{b},\ m_1,\ m_2$    complexity
$[\mathbf{S}\mathbf{A}^T] = \texttt{RP\_fun}(\mathbf{A}^T,\ m_1)$    $C(n, d, m_1)$
$[\mathbf{W}\mathbf{A}\mathbf{S}^T] = \texttt{RP\_fun}(\mathbf{A}\mathbf{S}^T,\ m_2)$    $C(n, m_1, m_2)$
$[\mathbf{\Sigma}_w, \mathbf{V}_w] = \texttt{svd}(\mathbf{W}\mathbf{A}\mathbf{S}^T,\ m_1)$    $O(m_2 m_1^2)$
$\tau = -\infty$, $i = -1$, $\boldsymbol{\nu}^0 = \mathbf{x}^0 = \mathbf{0}$, $\mathbf{z}^{0,0} = \mathbf{0}$
while until first stopping criteria do
    $i = i + 1$
    $\mathbf{h}^i = \mathbf{b} - \mathbf{A}\mathbf{x}^i$    $O(nd)$
    $\widetilde{\boldsymbol{\nu}}^i = \mathbf{S}\mathbf{A}^T\boldsymbol{\nu}^i$    (or $\widetilde{\boldsymbol{\nu}}^i = \mathbf{S}\mathbf{x}^i$)    $O(nm_1)$
    $\mathbf{z}^{i,0} = \mathbf{z}^{i-1,j+1}$, $j = -1$
    while until second stopping criteria do
        $j = j + 1$;
        $\mathbf{g}^{i,j} = \mathbf{V}_w^T\mathbf{S}\mathbf{A}^T(\mathbf{h}^i - \mathbf{A}\mathbf{S}^T\mathbf{z}^{i,j})$    $O(nm_1)$
        $\widetilde{\mathbf{z}}^{i,j} = \mathbf{V}_w^T(\mathbf{z}^{i,j} + \widetilde{\boldsymbol{\nu}}^i)$    $O(m_1^2)$
        $\mathbf{f}^{i,j} = \mathbf{\Sigma}_w^{-1}\mathbf{g}^{i,j} + \mathbf{\Sigma}_w\widetilde{\mathbf{z}}^{i,j}$    $O(m_1)$
        $\lambda_{i,j} = \underset{\lambda \geq \tau}{\operatorname{argmin}} \left\| \left(\mathbf{\Sigma}_w^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{f}^i \right\|_2 / \mathsf{tr}\left((\mathbf{\Sigma}_w^2 + \lambda\mathbf{I})^{-1}\right)$    $O(m_1)$
        $\Delta\mathbf{z}^{i,j} = \mathbf{V}_w\left(\mathbf{\Sigma}_w^2 + \lambda_{i,j}\mathbf{I}\right)^{-1}\left(\mathbf{g}^{i,j} - \lambda_{i,j}\widetilde{\mathbf{z}}^{i,j}\right)$    $O(m_1^2)$
        $\widehat{k} = m_1 - \lambda_{i,j}\mathsf{tr}\left(\left(\mathbf{\Sigma}_w^2 + \lambda_{i,j}\mathbf{I}\right)^{-1}\right)$
        $\beta_{1,j} = \widehat{k}/m_2$
        $\alpha_{1,j} = (1 - \beta_{1,j})^2$
        $\mathbf{z}^{i,j+1} = \mathbf{z}^{i,j} + \alpha_{1,j}\Delta\mathbf{z}^{i,j} + \beta_{1,j}(\mathbf{z}^{i,j} - \mathbf{z}^{i,j-1})$    $O(m_1)$
    end while
    $\Delta\boldsymbol{\nu}^i = (\mathbf{h}^i - \lambda_{i,j}\boldsymbol{\nu}^i - \mathbf{A}\mathbf{S}^T\mathbf{z}^{i,j+1})/\lambda_{i,j}$    $O(nm_1)$
    $\beta_{2,i} = \widehat{k}/m_1$
    $\alpha_{2,i} = (1 - \beta_{2,i})^2$
    $\boldsymbol{\nu}^{i+1} = \boldsymbol{\nu}^i + \alpha_{2,i}\Delta\boldsymbol{\nu}^i + \beta_{2,i}(\boldsymbol{\nu}^i - \boldsymbol{\nu}^{i-1})$    $O(d)$
    $\mathbf{x}^{i+1} = \mathbf{A}^T\boldsymbol{\nu}^{i+1}$    $O(nd)$
    $\tau = \max(\lambda_{i,j},\ \tau)$
end while
```

Similar to the `Hybrid M-IHS`, after obtaining $\lambda_{i,j}$ by minimizing $\mathbb{V}_3(\lambda)$ function, the momentum parameters for the inner iterations are set as

$$\beta_j = \mathsf{sd}_{\lambda_{i,j}}(\mathbf{\Sigma}_w)/m_2, \qquad \alpha_j = (1-\beta_j)^2.$$

After M inner iterations, the inexact dual step $\Delta\boldsymbol{\nu}^i$ is obtained by using the relation in eq. (4.22) and the momentum parameters for the outer loop are set as

$$\beta_i = \mathsf{sd}_{\lambda_{i,M}}(\mathbf{\Sigma}_w)/m_1, \qquad \alpha_i = (1-\beta_i)^2.$$

The risk functions given in eq. (4.27) and eq. (4.30) are equivalent to each other and both are used only for the derivation. An efficient form that requires $O(m_1)$ operations to compute $\mathbb{V}_3(\lambda)$ for a given λ is given in Algorithm 10.

For the problems with dense coefficient matrices, the computational complexity of the `Hybrid Primal Dual M-IHS` is $O(nd\log(m_1) + nm_1\log(m_2) + m_2m_1^2 + Nnd + NMnm_1)$, where m_1, m_2 are proportional to the prior estimate k' and the ROS sketching is used. The GKL bidiagonalization can be used to reduce the required operations by the SVD as shown in Algorithm 14. *Line 27* of Algorithm 10 can be incorporated into *Line 8* so that only one distributed matrix-vector multiplication with the coefficient matrix $\mathbf{A}$ is required per iteration. Similar to the highly over/under-determined cases, the `Hybrid Primal Dual M-IHS` computes the decomposition of $m_2 \times m_1$ dimensional matrix only once prior to the iterations. In sparse problems, by following the approaches used in the highly over/under-determined cases, the complexity of Algorithm 10 can be reduced to $O((NM + m_1)\mathrm{nnz}(\mathbf{A}) + m_1^3 + NMm_1^2)$. The total memory space required by Algorithm 10 is bounded by $O(nm_1 + m_2m_1)$.

4.3 Numerical Experiments and Comparisons

In this section, performance of the `Hybrid M-IHS` schemes will be compared with the direct and the deterministic hybrid methods through various numerical problems generated by using the IR Tool [11]. MATLAB implementation of the proposed techniques are given in the following link: `https://github.com/ibrahimkurban/Hybrid-M-IHS`.

4.3.1 Experiment setups

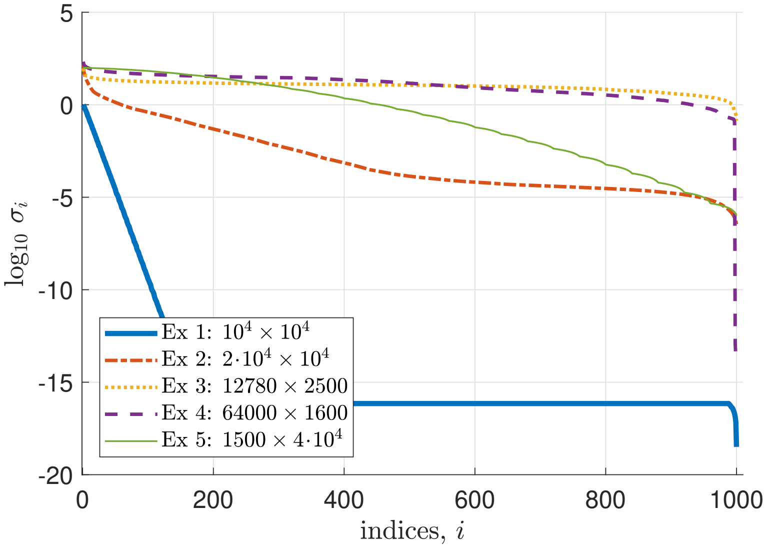


Figure 4.3: *The size and the singular value profiles of the coefficient matrices used in the numerical experiments.*

In the comparison, five difficult examples were used. The dimensions and the singular values of the coefficient matrices used in each example are given in fig. 4.3. The first two examples were used to test the `Hybrid Primal Dual M-IHS` on the square-like dimensional problems, the third and fourth examples were used to test the `Hybrid M-IHS` on the highly over-determined cases and the fifth example was used to test the `Hybrid Dual M-IHS` on the highly under-determined regimes. The data used in the first four examples were generated by using real life applications including image de-blurring, X-ray tomography and seismic travel-time tomography problems [11]; the fifth one was randomly generated. The details of data generation for each example are given below. In all examples, the additive i.i.d. Gaussian noise was used to model the error/noise vector $\mathbf{w}$ in eq. (2.1) and the techniques were tested at 8 different signal-to-noise ratio (SNR) in the range of 0.3% to 15%. At each SNR, the experiment was repeated for 20 different noise realizations and the results were averaged.

As for the error measure, instead of the true input $\mathbf{x}_0$, the *effective true input* $\mathbf{x}_{k^*} = \mathbf{V}_{k^*}\mathbf{V}_{k^*}^T\mathbf{x}_0$, where effective rank k^* is found by using the TSVD solution given in eq. (2.5), is used because, as discussed in Section 2.2.2, $\mathbf{x}_{k^*}$ represents the information about the true input $\mathbf{x}_0$ that can be extracted from the measurements. The effective rank k^*'s of the examples at each SNR are given in Table 4.1.

Table 4.1: *The average effective rank k^*'s of the examples at different SNR levels*

	0.3%	**0.6%**	**1%**	**4%**	**8%**	**10%**	**12%**	**15%**
Ex 1	293	259	245	195	163	164	162	158
Ex 2	1324	1006	759	417	261	224	188	176
Ex 3	2495	2489	2480	2460	2356	2306	2260	2106
Ex 4	1590	1581	1565	1473	1226	1221	1214	1180
Ex 5	879	832	791	679	603	579	563	527

Example 1 The first example is an image de-blurring problem. The *Gaussian* blur was used as the point spread function and the blurring level was set to highest rate, i.e., *severe*. A 100 × 100 dimensional image of the Hubble Space Telescope was used as the input and the *default* values were used for the rest of the problem specific parameters.

Example 2 The second example is a Seismic Travel-Time Tomography problem under the *Fresnel* wave model. A 100 × 100 dimensional random image with patterns of nonzero pixels, that is generated by *ppower* option, was used as the input and the *default* values were used for the rest of the parameters.

Example 3 The third example is an X-Ray Tomography problem with parallel beam geometry. A 50 × 50 dimensional *Shepp-Logan phantom* was used as the input image and the default values were used for the rest of the problem specific parameters.

Example 4 The fourth example is again a Seismic Travel-Time Tomography problem, but the waves are modeled as *straight* lines this time. The number of measurements was increased to obtain a highly over-determined problem. For this purpose, the number of *sources* and the *receivers* were set to 40 and 240, respectively. Also, the input image, which is generated by *ppower* option, was 40 × 40 dimensional.

Example 5 The fifth and the last example was synthetically by sampling the the entries of the coefficient matrix from the distribution $\mathcal{N}(1_d, \mathbf{\Gamma})$ where $\Gamma_{ij} = 6 \cdot 0.9^{|i-j|}$, providing highly correlated columns. Then its singular values were replaced with the *philips* singular value profile that is provided in RegTool [41]. The singular values were scaled for setting the condition number $\kappa(A)$ to 10^8. The entries of the input signal $\mathbf{x}_0$ were drawn uniformly from the interval of $[-0.5, 0.5]$ so that the coefficient matrix and input are independent from each other.

4.3.2 Compared methods and their implementation details

The first technique in the comparison set is the Oracle Regularized LS (*OR-LS*) solution $\mathbf{x}(\lambda^*)$ with parameter $\lambda^* = \text{argmin } \left\|\mathbf{x}(\lambda) - \mathbf{x}_{k^*}\right\|_2$. The *OR-LS* achieves the minimum error that can be obtained via the Tikhonov regularization, hence it serves as a theoretical lower bound for the errors produced by the other techniques in the comparison set. In practice, λ^* can be replaced by an estimate such as λ^{gcv} or λ_p^{gcv} which are the minimizers of the risk estimators given in eq. (2.8) and eq. (2.9), respectively. The resulting two solutions $\mathbf{x}(\lambda^{gcv})$ and $\mathbf{x}(\lambda_p^{gcv})$ are referred to as *GCV-full* and *GCV-partial* in the legends. The projection size

p in the *GCV-partial* was set to $\min(k^* + 500, n, d)$. Another technique in the comparison set is the Hybrid LSQR algorithm with WGCV risk estimator for which we used the MATLAB codes provided by Chung, et al in [51, 47]. For this algorithm, we tested two schemes: the first one *Hybrid LSQR (GCV-stop)* was obtained by terminating the iterations according to the GCV stopping criterion, and the second one *Hybrid LSQR (300)* was obtained by terminating after 300 iterations. In both schemes, the weight was selected through the adaptive weight selection technique described in [51] and the full re-orthogonalization step was applied in all iterations. As for the `Hybrid M-IHS` variants, ROS sketches, in which columns are sampled without replacement, was used with the Discrete Cosine Transform. For the first two examples, the pair $(m_1, m_2) = (2k^*,\ 5k^*)$, for the third and fourth examples $m = 2d$ and for the last example $m = 2n$ were used for the sketch sizes. During the bidiagonalization procedure for the `Hybrid M-IHS` variants, we applied re-orthogonalization only on $\mathbf{Q}_k$ matrix through the PROPACK package [50] as detailed in the Appendix C. Lastly, validity of the analysis given in Section 4.1 is demonstrated through the *Hybrid-modified* scheme that uses the GCV function given in eq. (4.2) for the Hybrid-LSQR algorithm with $L = \min(k^* + \ell, n, d)$ number of iterations for some ℓ, which are given in Table 4.2.

4.3.3 Obtained results

The error and parameter estimation results obtained in five examples described above are given in Figure 4.4, 4.5, 4.6, 4.7 and 4.8, respectively.

Average number of iterations that each iterative algorithm required to obtain the demonstrated results are given in Table 4.2. The results of the *GCV-partial* confirms that as long as the residual contains sufficient information about the noise statistics, i.e., projection dimension is sufficiently larger than k^*, the partial version of the GCV function given in eq. (2.9) can be used without degrading the performance of the risk estimator. Projecting the residual onto a smaller subspace of the range space of $\mathbf{A}$ might worsen the estimation because the projection

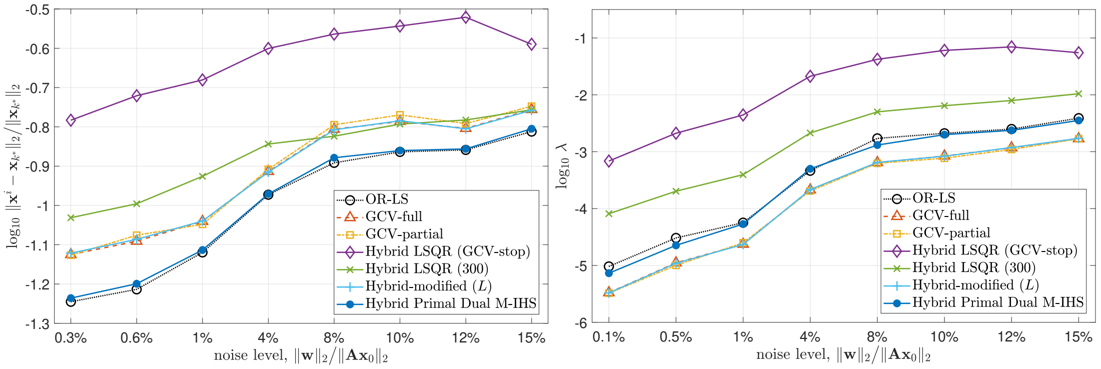


Figure 4.4: *Error and parameter estimation performances on Example 1 ($10^4 \times 10^4$)*

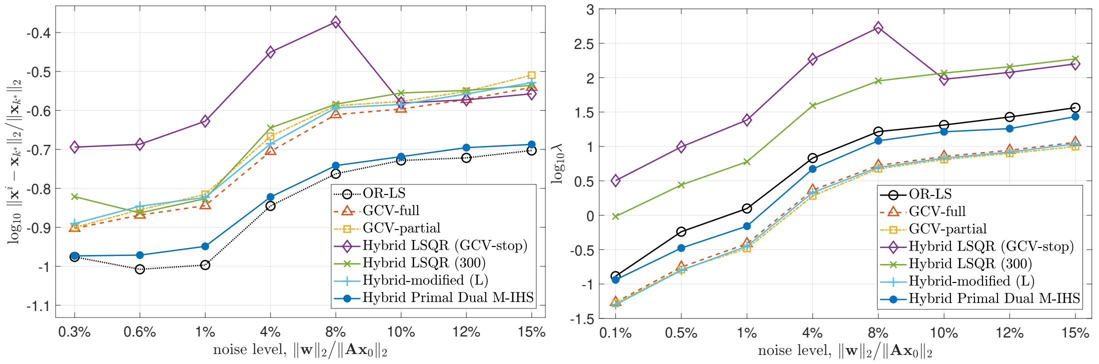


Figure 4.5: *Error and parameter estimation performances on Example 2 ($2{\cdot}10^4 \times 10^4$)*

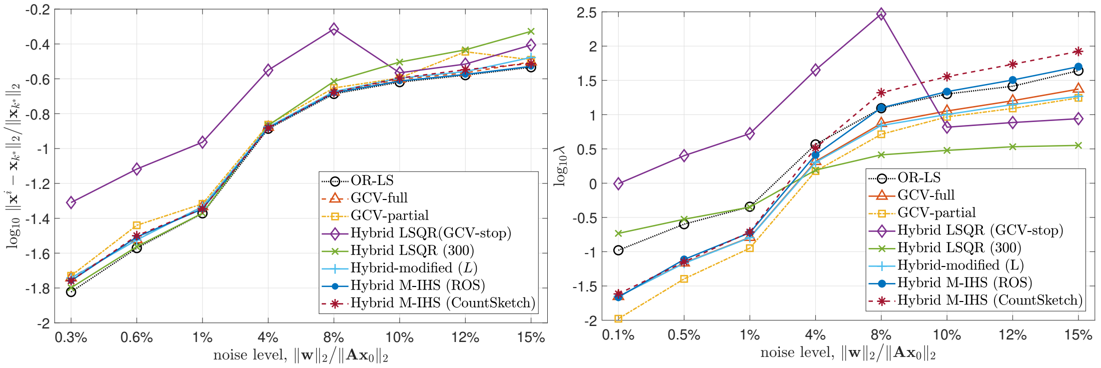


Figure 4.6: *Error and parameter estimation performances on Example 3* (12780×2500*)*

Table 4.2: *The number of iterations that the iterative algorithms need to obtain the results given in Figure 4.4, 4.5, 4.6, 4.7 and 4.8. While the* `Hybrid M-IHS` *variants require only one matrix-vector multiplications with the coefficient matrix per iteration, the number required by the GKL based hybrid methods significantly varies with respect to the preferred re-orthogonalization scheme.*

	Techniques	**0.3%**	**0.6%**	**1%**	**4%**	**8%**	**10%**	**12%**	**15%**
ex 1	Hybrid LSQR	39	27	23	8	4	4	3	38
	Hybrid-modified	593	559	545	495	463	464	462	458
	Hybrid M-IHS	14	15	14	11	13	12	12	10
ex 2	Hybrid LSQR	43	33	27	11	7	69	63	57
	Hybrid-modified	2260	1879	1676	1386	1266	1256	1227	1994
	Hybrid M-IHS	10	10	9	9	10	10	12	10
ex 3	Hybrid LSQR	38	29	22	9	6	133	124	126
	Hybrid-modified	2498	2492	2483	2463	2359	2309	2263	2109
	Hybrid M-IHS	18	17	16	13	12	9	10	9
ex 4	Hybrid LSQR	48	24	22	6	284	280	276	256
	Hybrid-modified	1600	1600	1600	1600	1600	1600	1593	1534
	Hybrid M-IHS	18	18	17	13	10	8	9	8
ex 5	Hybrid LSQR	177	109	58	17	10	98	82	70
	Hybrid-modified	1179	1132	1091	979	903	879	863	827
	Hybrid M-IHS	7	7	7	8	10	10	10	10

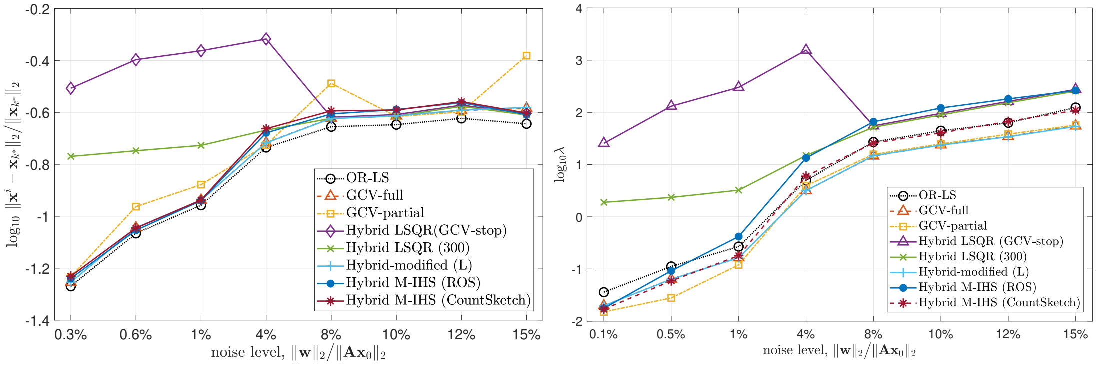


Figure 4.7: *Error and parameter estimation performances on Example 4* (64000×1600)

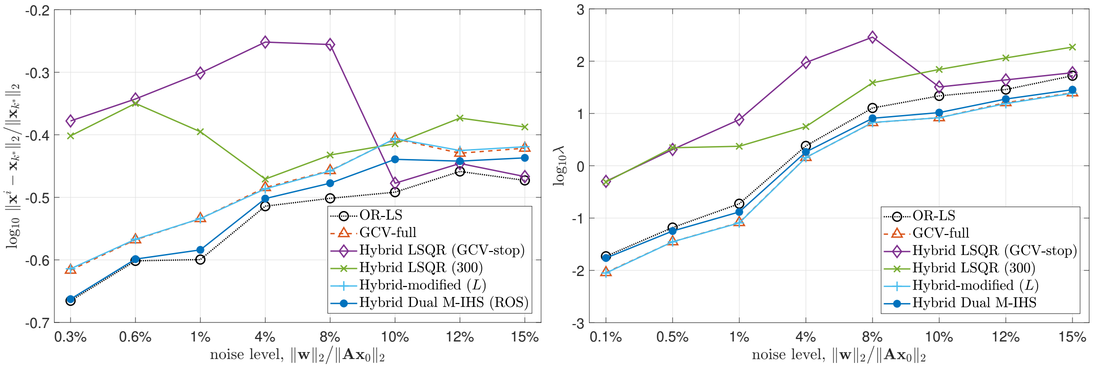
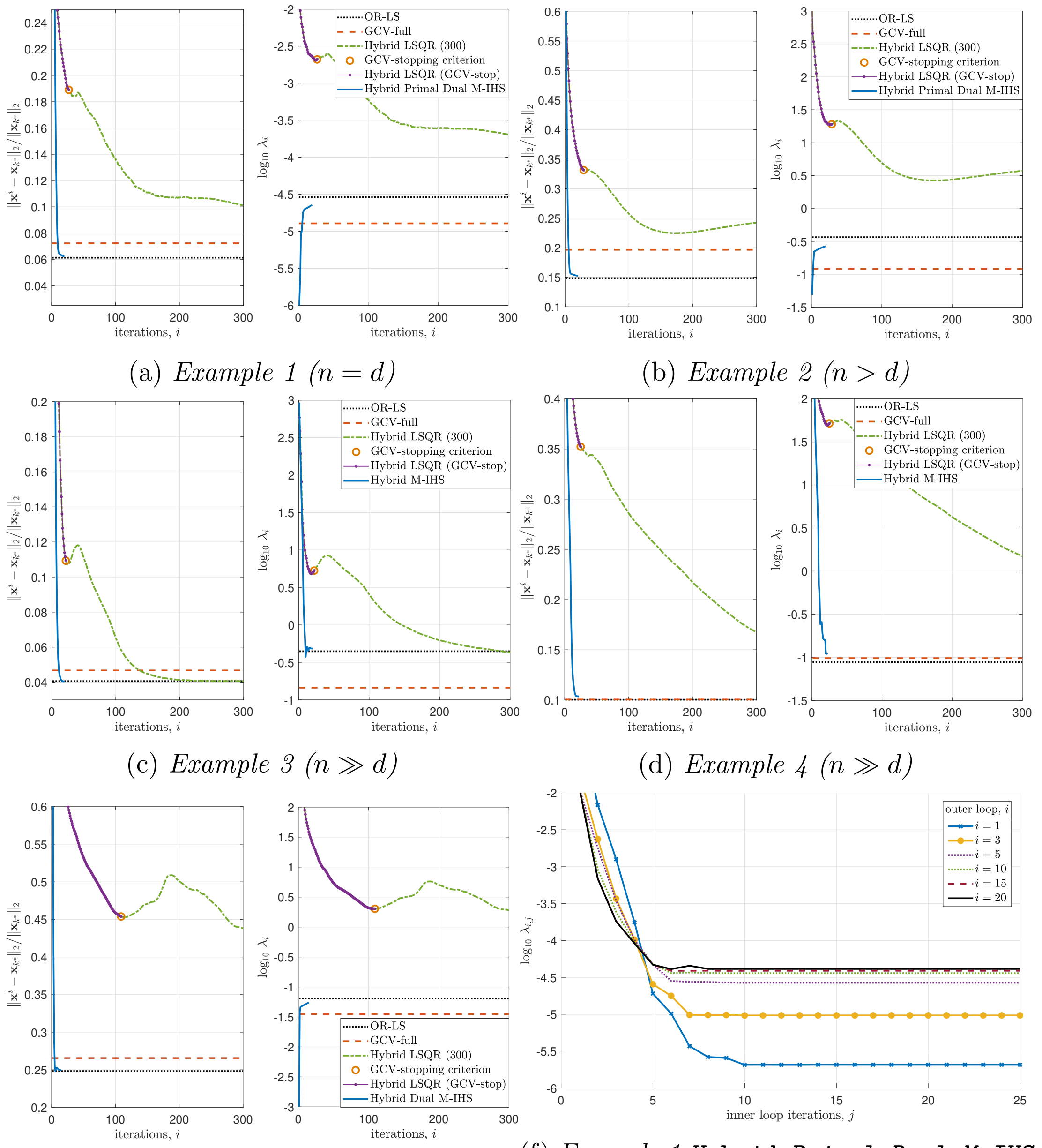


Figure 4.8: *Error and parameter estimation performances on Example 5* ($1500 \times 4{\cdot}10^4$)

discards the residual components that contain useful statistics about the noise, but still the partial version generates satisfactory results compared to the naive GCV.

While the *GCV* variants consistently underestimate the optimal regularization parameter, the *Hybrid LSQR* techniques overestimate the λ^* value because the GCV stopping criterion typically terminates the iterations too early. When the number of iterations for the *Hybrid LSQR* technique is increased up to 300, the parameter estimation gets better as demonstrated through the *Hybrid GCV (300)* scheme, but still 300 iterations are not enough for most of the cases to obtain satisfactory results. If the correction given in eq. (4.2) is used with a number of iterations that sufficiently exceeds the effective rank of the problem, such as L, then to obtain stable results with the conventional hybrid methods seems possible as seen on the *Hybrid-modified* scheme. However, although *Hybrid-modified* generates stable results, its time complexity is very high due to number



(a) *Example 1 ($n = d$)*

(b) *Example 2 ($n > d$)*

(c) *Example 3 ($n \gg d$)*

(d) *Example 4 ($n \gg d$)*

(e) *Example 5 ($n \ll d$)*

(f) *Example 1* **Hybrid Primal Dual M-IHS** *iterations*

Figure 4.9: *Convergence behaviour of the hybrid methods. The GCV stopping criterion causes the Hybrid LSQR algorithm to terminate too early, but even 300 iterations of the Hybrid LSQR is not sufficient to obtain an accuracy that is provided by the direct methods. On the other side, the proposed* **Hybrid M-HS** *variants converge quickly. Plot* ***(f)*** *shows that* **Hybrid Primal Dual M-IHS** *inherits the convergence behaviour of the* **Hybrid Dual M-IHS** *in the outer loop, i.e., λ_i's starts small and get larger through the iterations while inherits the convergence behaviour of the* **Hybrid M-IHS** *in the inner loop iterations, i.e., $\lambda_{i,j}$ starts large and get smaller.*

of matrix-vector multiplications with $\mathbf{A}$ and the re-orthogonalization steps in the bidiagonalization procedure.

On the other hand, the proposed `Hybrid M-IHS` variants are capable of producing close estimates to λ^* in a steady manner in all dimension regimes and at each SNR. In example 3 and 4, the sparsity ratio of the coefficient matrices are approximately 1.8% and 2.9% respectively. Thus in Figure 4.6 and Figure 4.7, we also show the performance of the `Hybrid M-IHS` technique when the sparse subspace embeddings are used as the sketching matrix. The Count sketch contains only one non-zero element in each column and therefore can be applied to the coefficient matrix in $O(\text{nnz}(\mathbf{A}))$ operations. In these experiments, the Count sketch was used with the same sketch size as the ROS matrices, i.e., $m = 2d$. One of the main motivation behind the proposed `Hybrid M-IHS` techniques was to decrease the number of matrix-vector multiplications with $\mathbf{A}$ by reducing the number of iterations. As shown on the Table 4.2, in all conducted experiments, the total number of iterations does not exceed a few dozens for the `Hybrid M-IHS` variants to achieve comparable accuracies with the direct methods that are applied on the full data. Conversely, the number of iterations required by the conventional hybrid methods changes significantly with respect to the spectral properties of $\mathbf{A}$. In Figure 4.9, at an SNR level of 1%, we demonstrate the convergence behaviour of the Hybrid-LSQR and the `Hybrid M-IHS` techniques by drawing the error and the parameter estimation progress over all iterations. Note that for (m_1, m_2) values, the prior k' is assumed to be k^* which is the smallest possible value. For practical values of k', the sketch (m_1, m_2) would be larger, hence the convergence of the `Hybrid Primal Dual M-IHS` will be faster.

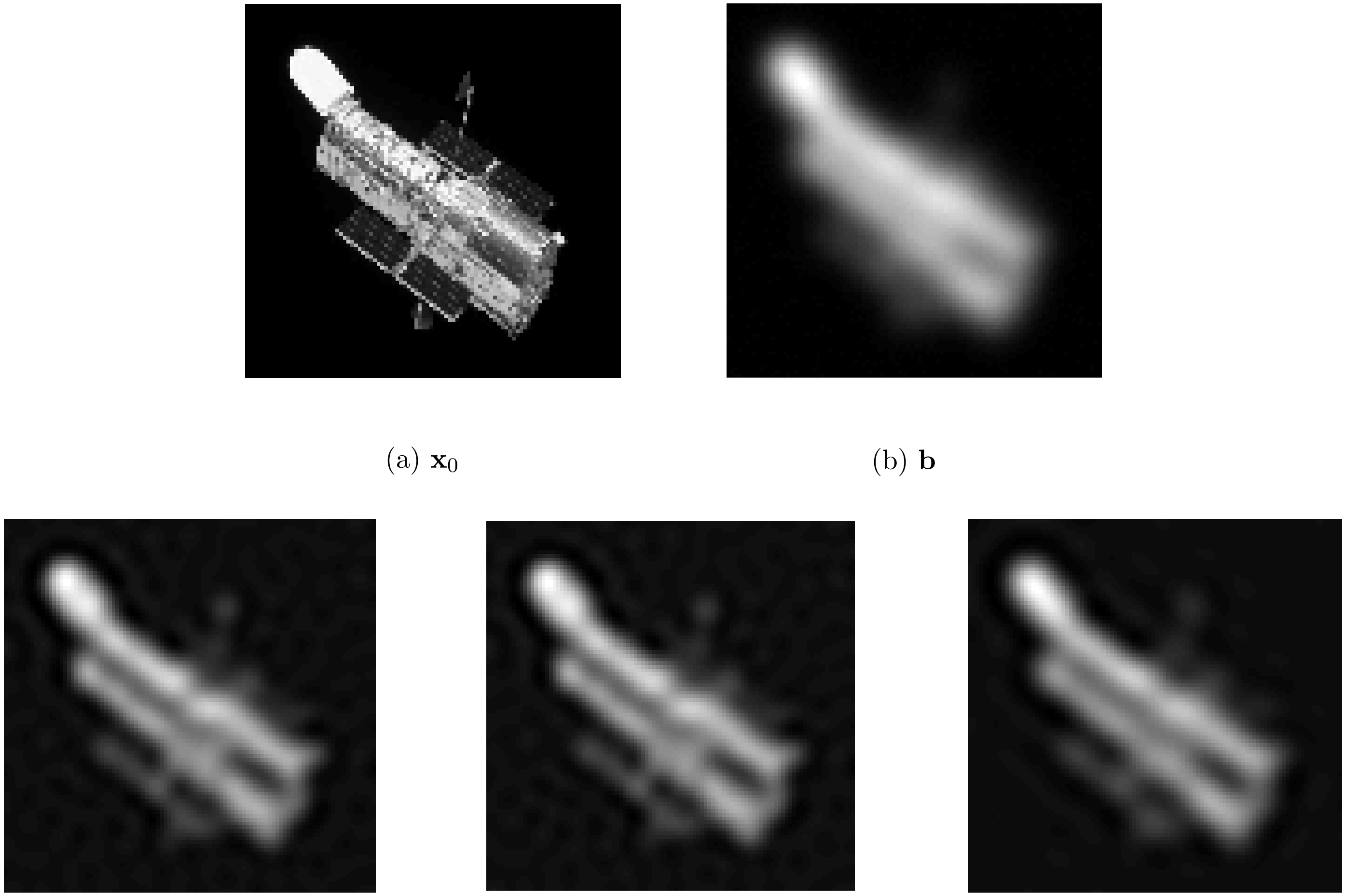
For a qualitative comparison of the reconstructed solutions that are obtained at an SNR level of 1%, we demonstrated the true input, the noisy measurements and the reconstructed signals of the first four examples in Figure 4.10, 4.11, 4.12 and 4.13, respectively. The images reconstructed by using the *Hybrid LSQR (GCV)* scheme are highly over-smoothed whereas the proposed `Hybrid M-IHS` variants reconstructs images that are almost identical to the results of the *OR-LS* scheme. In Table 4.3, for a quantitative comparison, we give the the Peak-SNR (PSNR) values of the reconstructed images at three different SNR levels. The PSNR values are calculated by taking the effective true input image as the reference image to emphasize the performance of the algorithms on the recoverable parts of the solutions. In all examples and SNR's, unlike *Hybrid LSQR (GCV)*, the reconstructions of the `M-IHS` variants have PSNR values that are very close to the ones obtained by the *OR-LS* scheme.

Table 4.3: *PSNR (in dB) values of the reconstructed images measured with respect to the effective true input* $\mathbf{x}_{k^*}$.

ex. no	**ex. 1**			**ex. 2**			**ex. 3**			**ex. 4**		
$\lVert\mathbf{w}\rVert/\lVert\mathbf{A}\mathbf{x}_0\rVert$	0.3%	1%	10%	0.3%	1%	10%	0.3%	1%	10%	0.3%	1%	10%
OR-LS	36.00	35.71	31.48	28.46	23.65	22.89	49.20	39.79	24.93	35.92	28.99	22.56
Hybrid M-IHS	35.95	35.6	29.27	28.44	23.60	22.89	47.70	39.49	24.82	36.02	28.92	22.56
Hybrid LSQR	30.57	29.80	24.93	22.58	16.09	19.37	38.40	31.43	24.02	15.95	15.93	22.04

4.4 Contributions and Conclusion

In this chapter, we proposed a group of novel hybrid schemes adapted for the randomized preconditioning techniques to estimate the regularization parameter λ for the LS problem given in eq. (2.3) by using the lower dimensional sub-problems, which are constructed by random projections during the iterations, without accessing the full data pair except for the gradient computations. The regularization parameter λ is estimated along with the iterations and the corresponding regularized solution is produced as a result. Since, as the core solver of the proposed hybrid scheme, we choose to use the `M-IHS` techniques that offer several advantageous properties for parallel or distributed memory environments prevalent in large scale applications [37], the `Hybrid M-IHS` schemes estimate regularization parameters from the lower dimensional sub-problems that are solved for determining the approximate Newton step, that will be referred to as the Hessian Sketching (HS) step. A proxy of the Generalized Cross Validation (GCV) is used in the estimation of the regularization parameter λ. Although the individual sub-problems constitute crude approximations for the full LS problem, the accuracy of their solutions increases exponentially over the iterations [55] and hence, accuracy of the λ estimates converge rapidly to a proper value for the full problem. The number of multiplications with the coefficient matrix, which is the main complexity issue of the conventional deterministic hybrid methods, is substantially reduced by the proposed `Hybrid M-IHS` schemes. We demonstrated the performance of the `Hybrid M-IHS` on several realistic problems extracted from IR tool [11]. In all the conducted experiments, the `Hybrid M-IHS` techniques consistently produce better error results than the direct methods by accessing the full data by a significantly fewer number of times than the conventional hybrid techniques.



(a) $\mathbf{x}_0$

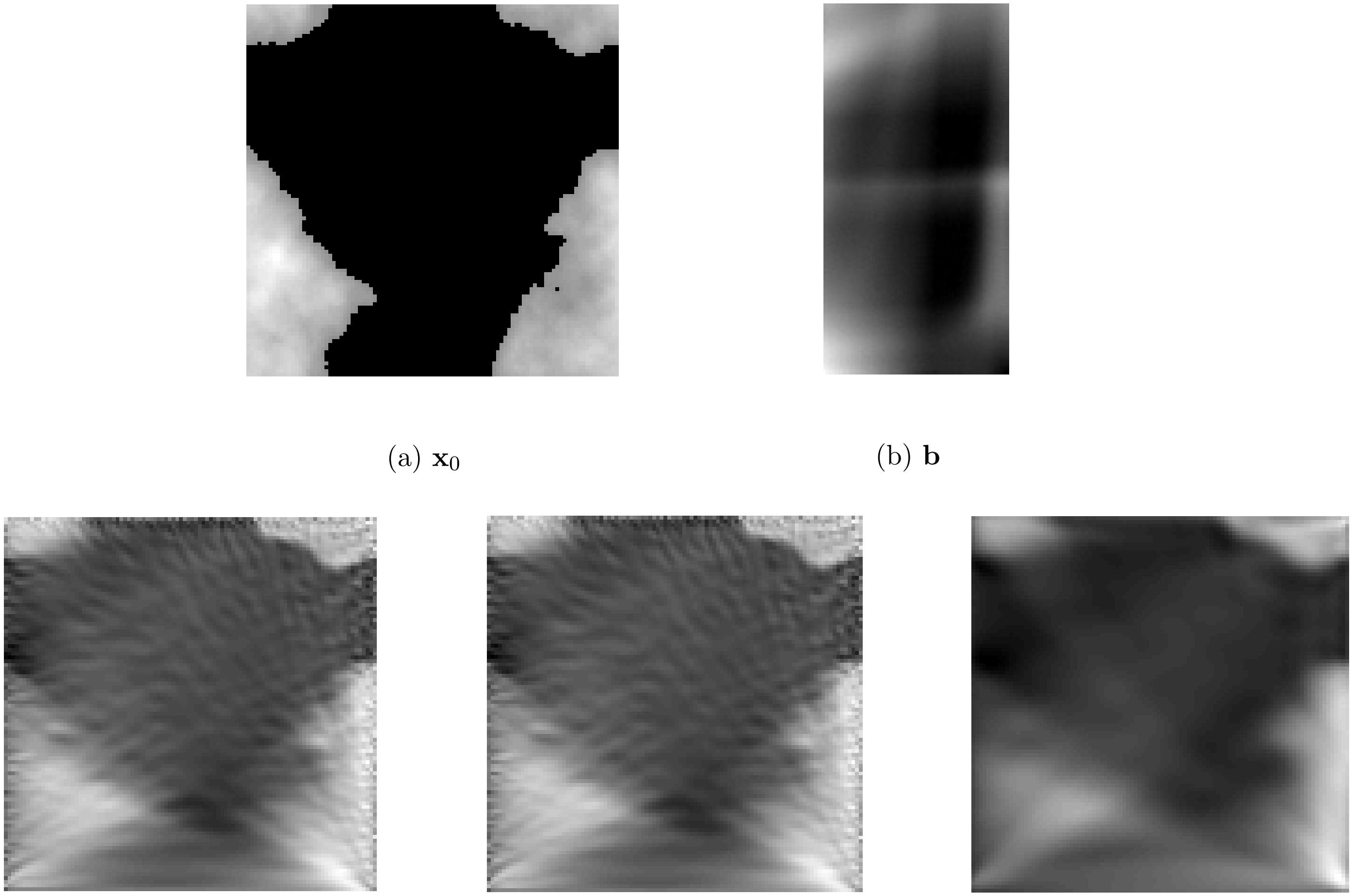
(b) $\mathbf{b}$

(c) $\mathbf{x}_{\mathit{Oracle}}$

(d) $\mathbf{x}_{\mathit{M\text{-}IHS}}$

(e) $\mathbf{x}_{\mathit{Hybrid\text{-}LSQR}}$

Figure 4.10: *Example 1 ($n = d$): deblurring problem with Gaussian psf function*



(a) $\mathbf{x}_0$

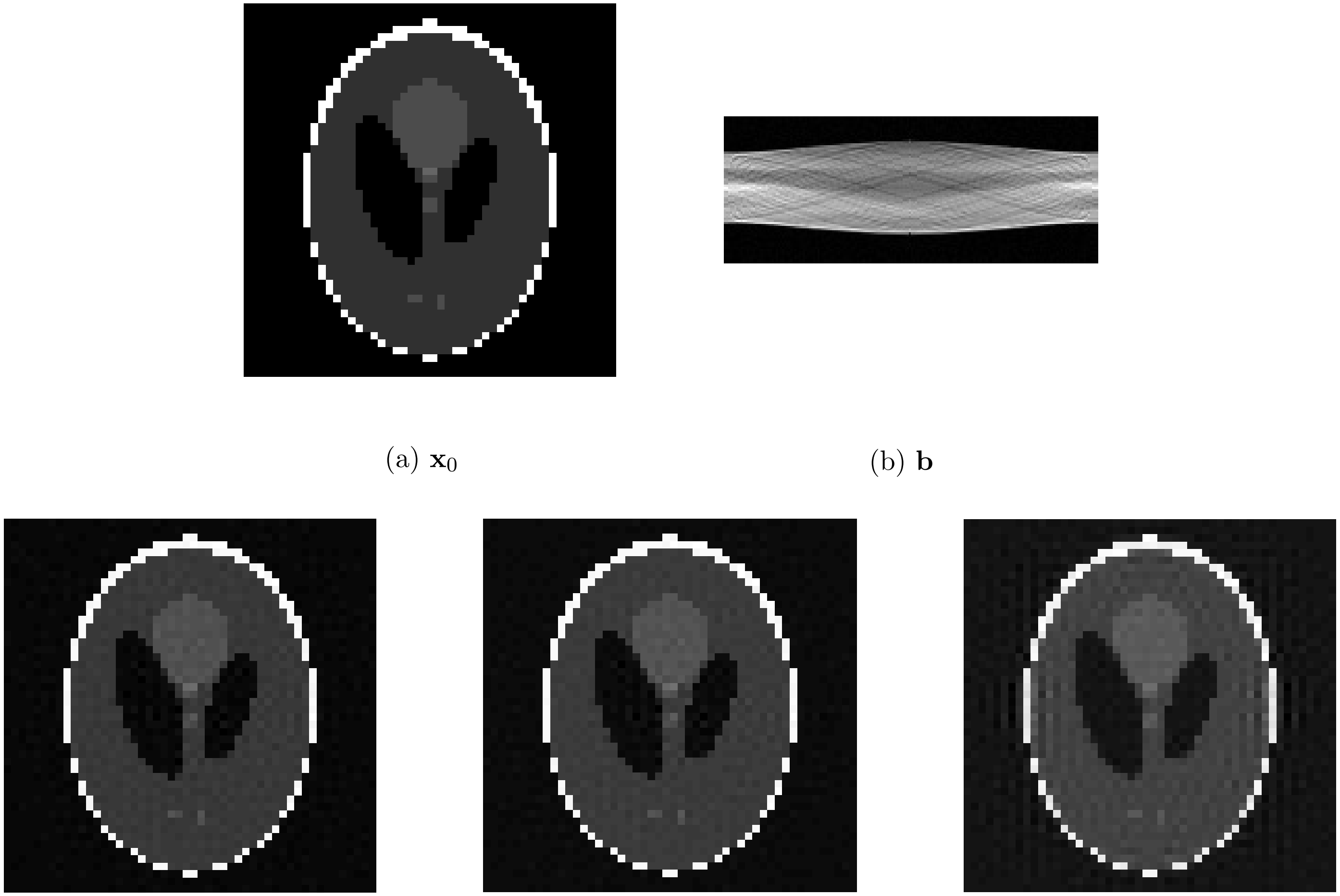
(b) $\mathbf{b}$

(c) $\mathbf{x}_{\mathtt{Oracle}}$

(d) $\mathbf{x}_{\mathtt{M\text{-}IHS}}$

(e) $\mathbf{x}_{\mathtt{Hybrid\text{-}LSQR}}$

Figure 4.11: *Example 2 ($n \geq d$): seismic travel-time tomography problem with Fresnel wave model*



(a) $\mathbf{x}_0$

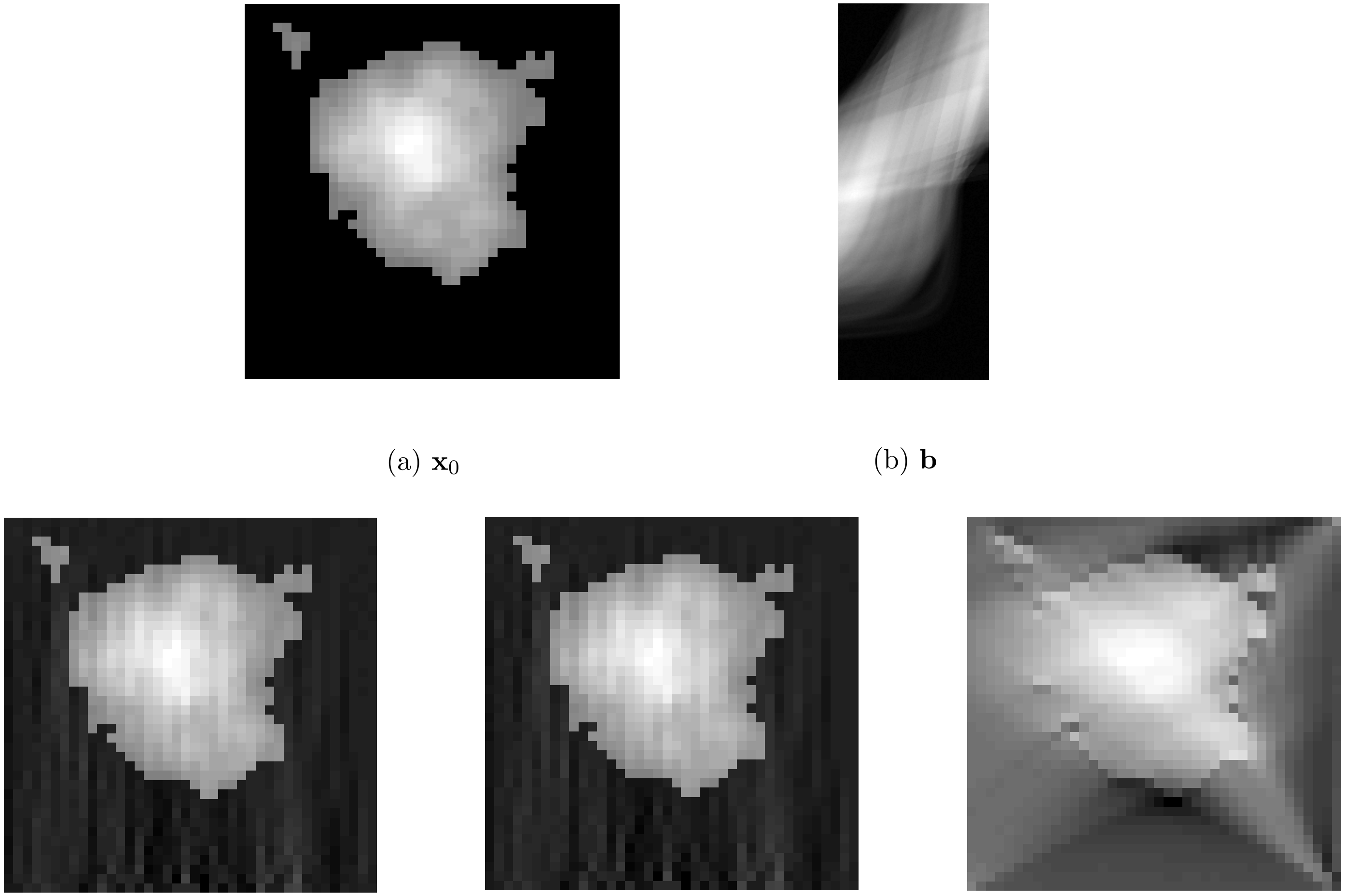
(b) $\mathbf{b}$

(c) $\mathbf{x}_{\mathtt{Oracle}}$

(d) $\mathbf{x}_{\mathtt{M\text{-}IHS}}$

(e) $\mathbf{x}_{\mathtt{Hybrid\text{-}LSQR}}$

Figure 4.12: *Example 3 ($n \gg d$): X-ray tomography problem*



(a) $\mathbf{x}_0$

(b) $\mathbf{b}$

(c) $\mathbf{x}_{\mathtt{Oracle}}$

(d) $\mathbf{x}_{\mathtt{M\text{-}IHS}}$

(e) $\mathbf{x}_{\mathtt{Hybrid\text{-}LSQR}}$

Figure 4.13: *Example 4 ($n \gg d$): seismic travel-time tomography problem with straight line model*

Chapter 5

Conclusions and Future Work

In this thesis, we presented a group solvers for large scale linear LS problems. In the first part, $\ell 2$-norm regularization parameter is assumed to be either zero or a known value. Then, a novel randomized solver `M-IHS` which is an improvement of the Iterative Hessian Sketch and the randomized preconditioning is proposed and its convergence properties are analyzed in detail. For the un-regularized LS problems, the `M-IHS` solver is efficient only in highly over-determined problems. Our asymptotic analysis reveals that its convergence rate is independent of the spectral properties of the coefficient matrix and is determined by the ratio between the number of columns in the coefficient matrix and the sketch size which is controlled by the user. For the regularized LS problems, the idea of the `M-IHS` solver is extended to the highly under-determined problems and the `Dual M-IHS` solver is obtained. Our non-asymptotic analyses show that in the regularized LS problems, the sketch size can be chosen proportional to the statistical dimension for the `M-IHS` and the `Dual M-IHS` to converge at an exponential rate with constant probability, which means that the `M-IHS` variants can be used for any regularized LS problems as long as the statistical dimension is sufficiently smaller than at least one dimension of the coefficient matrix. We also showed that the exponential rate is empirically determined by the ratio between the statistical dimension and the sketch size, which confirms the asymptotic analysis made for the un-regularized case. In the light of these theoretical findings, we also derived the

`Primal Dual M-IHS` that applies two-stage sketching on the coefficient matrix to gain further computational savings.

The `M-IHS` variants offer several advantageous properties for modern computation environments such as parallel or distributed memory systems that are prevalent in large scale applications. First of all, when sub-solvers are utilized in the iterations, the `M-IHS` variants do not require any matrix decomposition or inversion, unlike well-known randomized preconditioning methods such as the Blendenpik and the LSRN. Therefore, the `M-IHS` variants might be the methods of choice in large scale problems such as the 3D imaging where even the decomposition of the sketched matrix is not feasible to compute. Secondly, the `M-IHS` variants do not require any inner product or norm calculations in the iterations, hence avoid synchronization steps in parallel computing, which results in overwhelming advantages over the CG or the GMRES like iterative solvers in distributed or hierarchical memory systems. Moreover, the `M-IHS` variants can be used for the problems where the coefficient matrix is sparse or an operator that allows matrix-vector multiplications.

In the second part, the main focus is on the estimation of $\ell 2$-norm regularization parameter. We introduced hybrid schemes for the `M-IHS` variants to estimate the regularization parameter along with the iterations. Unlike conventional hybrid methods, the proposed hybrid schemes are based on random projections instead of deterministic projections onto the Krylov Subspaces. In the `Hybrid M-IHS` variants, the regularization parameter is estimated from the sub-problems that arise during the iterations of the `M-IHS` variants by using a proxy of the GCV technique. Although individual sub-problems constitute crude approximations for the full problem, when their solutions are built on the top of each other, they eventually allow estimation of accurate parameters. Thus, the parameter sequence estimated from these lower dimensional sub-problems rapidly converge to a proper regularization parameter for the full problem. In various experiments conducted over realistic problems such as image de-blurring, X-ray tomography and seismic travel-time tomography, the `Hybrid M-IHS` variants consistently estimate better regularization parameters at all SNR levels than the conventional hybrid methods, hence allow less error in the reconstructions. Moreover, to get these results, `M-IHS` variants require significantly fewer number of matrix-vector multiplications with the coefficient matrix. Therefore, unlike the conventional hybrid methods, as long as the decomposition of the sketched matrix can be computed efficiently, the proposed hybrid schemes are not only effective in the sequential systems, but they can also be successfully adapted into the parallel or distributed memory environments.

The effect of the inexact sub-solvers on the convergence rate of the `M-IHS` algorithms can be studied as a future direction. Such analyses would reveal the target accuracy for the solutions of the sub-systems and the maximum number of iterations. Similar to the conventional hybrid methods, the `Hybrid M-IHS` techniques do not have a guarantee for the selected regularization parameters since the estimation of the parameter is based on the GCV heuristics. Any advancement in certifying the selected parameter would be seminal in the literature of the hybrid methods. Lastly, all the techniques discussed or proposed in this thesis require more than one accesses to the coefficient matrix, but for many contemporary applications, even one access to the coefficient matrix is hardly possible. In such problems, the classical sketching techniques can be investigated to estimate robust regularization parameters and to construct accurate solutions.

Bibliography

[1] J. Wang, J. D. Lee, M. Mahdavi, M. Kolar, N. Srebro, *et al.*, "Sketching meets random projection in the dual: A provable recovery algorithm for big and high-dimensional data," *Electron. J. Stat.*, vol. 11, no. 2, pp. 4896–4944, 2017.

[2] J. Christensen and G. Gustafson, "A brief history of linear algebra," 2012.

[3] R. Hart, *The Chinese roots of linear algebra.* JHU Press, 2011.

[4] H. H. Goldstine, *A History of Numerical Analysis from the 16th through the 19th Century*, vol. 2. Springer Science & Business Media, 2012.

[5] G. Stewart, "The decompositional approach to matrix computation," *Comput. Sci. Eng.*, vol. 2, no. 1, pp. 50–59, 2000.

[6] H. A. Van Der Vorst, "Krylov subspace iteration," *Comput. Sci. Eng.*, vol. 2, no. 1, pp. 32–37, 2000.

[7] G. Golub and C. Van Loan, *Matrix Computations.* Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2013.

[8] A. Greenbaum, *Iterative methods for solving linear systems*, vol. 17. Siam, 1997.

[9] M. Benzi, "Preconditioning techniques for large linear systems: a survey," *J. Comput. Phys.*, vol. 182, no. 2, pp. 418–477, 2002.

[10] M. E. Kilmer and D. P. O'Leary, "Choosing regularization parameters in iterative methods for ill-posed problems," *SIAM J. Matrix Anal. Appl.*, vol. 22, no. 4, pp. 1204–1221, 2001.

[11] S. Gazzola, P. C. Hansen, and J. G. Nagy, "Ir tools: a matlab package of iterative regularization methods and large-scale test problems," *Numer. Algorithms*, vol. 81, no. 3, pp. 773–811, 2019.

[12] L. Shi, L. Zhao, W.-Z. Song, G. Kamath, Y. Wu, and X. Liu, "Distributed least-squares iterative methods in networks: A survey," *arXiv preprint arXiv:1706.07098*, 2017.

[13] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods*, vol. 43. SIAM, 1994.

[14] M. Pilanci and M. J. Wainwright, "Randomized sketches of convex programs with sharp guarantees," *IEEE Trans. Inform. Theory*, vol. 61, no. 9, pp. 5096–5115, 2015.

[15] M. W. Mahoney *et al.*, "Randomized algorithms for matrices and data," *Found. Trends Mach. Learn.*, vol. 3, no. 2, pp. 123–224, 2011.

[16] D. P. Woodruff *et al.*, "Sketching as a tool for numerical linear algebra," *Found. Trends Theor. Comput. Sci.*, vol. 10, no. 1–2, pp. 1–157, 2014.

[17] M. Pilanci, *Fast randomized algorithms for convex optimization and statistical estimation.* PhD thesis, UC Berkeley, 2016.

[18] Å. Björck, *Numerical methods for least squares problems.* Philadelphia, PA, USA: SIAM, 1996.

[19] P. C. Hansen, *Discrete inverse problems: insight and algorithms*, vol. 7. SIAM, 2010.

[20] D. L. Donoho *et al.*, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[21] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[22] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[24] S. Bubeck *et al.*, “Convex optimization: Algorithms and complexity,” *Found. Trends Mach. Learn.*, vol. 8, no. 3-4, pp. 231–357, 2015.

[25] M. Pilanci and M. J. Wainwright, “Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence,” *SIAM J. Optim.*, vol. 27, no. 1, pp. 205–245, 2017.

[26] D. G. Luenberger, *Introduction to linear and nonlinear programming*, vol. 28. Addison-Wesley Reading, MA, 1973.

[27] B. Bartan and M. Pilanci, “Straggler resilient serverless computing based on polar codes,” in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 276–283, IEEE, 2019.

[28] E. Jonas, Q. Pu, S. Venkataraman, I. Stoica, and B. Recht, “Occupy the cloud: Distributed computing for the 99%,” in *Proceedings of the 2017 Symposium on Cloud Computing*, pp. 445–451, ACM, 2017.

[29] X. Meng, M. A. Saunders, and M. W. Mahoney, “Lsrn: A parallel iterative solver for strongly over-or underdetermined systems,” *SIAM J. Sci. Comput.*, vol. 36, no. 2, pp. C95–C118, 2014.

[30] M. H. Gutknecht and S. Röllin, “The chebyshev iteration revisited,” *Parallel Comput.*, vol. 28, no. 2, pp. 263–283, 2002.

[31] C. W. Groetsch, “Integral equations of the first kind, inverse problems and regularization: a crash course,” in *Journal of Physics: Conference Series*, vol. 73, p. 012001, IOP Publishing, 2007.

[32] P. C. Hansen, “The discrete picard condition for discrete ill-posed problems,” *BIT*, vol. 30, no. 4, pp. 658–672, 1990.

[33] C. R. Vogel, *Computational methods for inverse problems*, vol. 23. SIAM, 2002.

[34] T. K. Moon and W. C. Stirling, *Mathematical methods and algorithms for signal processing*, vol. 1. Prentice hall Upper Saddle River, NJ, 2000.

[35] H. Avron, K. L. Clarkson, and D. P. Woodruff, “Faster kernel ridge regression using sketching and preconditioning,” *SIAM J. Matrix Anal. Appl.*, vol. 38, no. 4, pp. 1116–1138, 2017.

[36] H. Avron, K. L. Clarkson, and D. P. Woodruff, “Sharper bounds for regularized data fitting,” *arXiv preprint arXiv:1611.03225*, 2016.

[37] I. K. Ozaslan, M. Pilanci, and O. Arikan, “Regularized momentum iterative hessian sketch for large scale linear system of equations,” *arXiv preprint arXiv:1912.03514*, 2019.

[38] P. C. Hansen, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, vol. 4. SIAM, 2005.

[39] Y. Huang and Z. Jia, “Some results on the regularization of lsqr for large-scale discrete ill-posed problems,” *Sci. China Math.*, vol. 60, no. 4, pp. 701–718, 2017.

[40] R. A. Renaut, S. Vatankhah, and V. E. Ardestani, “Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted gcv for projected systems,” *SIAM J. Sci. Comput.*, vol. 39, no. 2, pp. B221–B243, 2017.

[41] P. C. Hansen, “Regularization tools: a matlab package for analysis and solution of discrete ill-posed problems,” *Numer. Algorithms*, vol. 6, no. 1, pp. 1–35, 1994.

[42] F. Lucka, K. Proksch, C. Brune, N. Bissantz, M. Burger, H. Dette, and F. Wübbeling, “Risk estimators for choosing regularization parameters in ill-posed problems-properties and limitations,” *Inverse Probl. Imaging*, vol. 12, no. 5, pp. 1121–1155, 2018.

[43] Y. C. Eldar, “Generalized sure for exponential families: Applications to regularization,” *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 471–481, 2008.

[44] G. H. Golub, M. Heath, and G. Wahba, “Generalized cross-validation as a method for choosing a good ridge parameter,” *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[45] P. C. Hansen and D. P. O’Leary, “The use of the l-curve in the regularization of discrete ill-posed problems,” *SIAM J. Sci. Comput.*, vol. 14, no. 6, pp. 1487–1503, 1993.

[46] I. Hnětynková, M. Plešinger, and Z. Strakoš, “The regularizing effect of the golub-kahan iterative bidiagonalization and revealing the noise level in the data,” *BIT*, vol. 49, no. 4, pp. 669–696, 2009.

[47] J. Chung and K. Palmer, “A hybrid lsmr algorithm for large-scale tikhonov regularization,” *SIAM J. Sci. Comput.*, vol. 37, no. 5, pp. S562–S580, 2015.

[48] J. Chung and A. K. Saibaba, “Generalized hybrid iterative methods for large-scale bayesian inverse problems,” *SIAM J. Sci. Comput.*, vol. 39, no. 5, pp. S24–S46, 2017.

[49] C. C. Paige and M. A. Saunders, “Lsqr: An algorithm for sparse linear equations and sparse least squares,” *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, 1982.

[50] R. Larsen, “Lanczos bidiagonalization with partial reorthogonalization,” *DAIMI Report Series*, vol. 27, Dec. 1998.

[51] J. Chung, J. G. Nagy, and D. P. O’Leary, “A weighted gcv method for lanczos hybrid regularization,” *Electronic Transactions on Numerical Analysis*, vol. 28, no. Electronic Transactions on Numerical Analysis, 2008.

[52] T. Sarlos, "Improved approximation algorithms for large matrices via random projections," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 143–152, IEEE, 2006.

[53] D. M. Kane and J. Nelson, "Sparser johnson-lindenstrauss transforms," *J. ACM*, vol. 61, no. 1, p. 4, 2014.

[54] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, "Faster least squares approximation," *Numer. Math*, vol. 117, no. 2, pp. 219–249, 2011.

[55] M. Pilanci and M. J. Wainwright, "Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1842–1879, 2016.

[56] V. Rokhlin and M. Tygert, "A fast randomized algorithm for overdetermined linear least-squares regression," *Proc. Natl. Acad. Sci. USA*, vol. 105, no. 36, pp. 13212–13217, 2008.

[57] H. Avron, P. Maymounkov, and S. Toledo, "Blendenpik: Supercharging lapack's least-squares solver," *SIAM J. Sci. Comput.*, vol. 32, no. 3, pp. 1217–1236, 2010.

[58] M. Bertero and M. Piana, "Inverse problems in biomedical imaging: modeling and methods of solution," in *Complex systems in biomedicine*, pp. 1–33, Springer, 2006.

[59] A. Chambolle, "Continuous optimization, an introduction," 2016.

[60] I. K. Ozaslan, M. Pilanci, and O. Arikan, "Iterative hessian sketch with momentum," *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[61] B. Recht, "Cs726-lyapunov analysis and the heavy ball method," 2010.

[62] R. J. Muirhead, *Aspects of multivariate statistical theory*, vol. 197. John Wiley & Sons, 2009.

[63] A. Edelman and Y. Wang, "Random matrix theory and its innovative applications," in *Advances in Applied Mathematics, Modeling, and Computational Science*, pp. 91–116, Springer, 2013.

[64] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint arXiv:1011.3027*, 2010.

[65] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, 1964.

[66] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[67] B. O'donoghue and E. Candes, "Adaptive restart for accelerated gradient schemes," *Found. Comput. Math.*, vol. 15, no. 3, pp. 715–732, 2015.

[68] J. Nelson and H. L. Nguyên, "Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 117–126, IEEE, 2013.

[69] M. B. Cohen, J. Nelson, and D. P. Woodruff, "Optimal approximate matrix product in terms of stable rank," *arXiv preprint arXiv:1507.02268*, 2015.

[70] M. Thorup and Y. Zhang, "Tabulation-based 5-independent hashing with applications to linear probing and second moment estimation," *SIAM J. Comput.*, vol. 41, no. 2, pp. 293–331, 2012.

[71] M. B. Cohen, "Nearly tight oblivious subspace embeddings by trace inequalities," in *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pp. 278–287, SIAM, 2016.

[72] J. Nelson and H. L. NguyÅn, "Lower bounds for oblivious subspace embeddings," in *International Colloquium on Automata, Languages, and Programming*, pp. 883–894, Springer, 2014.

[73] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemp. Math.*, vol. 26, no. 189-206, p. 1, 1984.

[74] D. Kane, R. Meka, and J. Nelson, “Almost optimal explicit johnson-lindenstrauss families,” in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 628–639, Springer, 2011.

[75] J. Nocedal and S. Wright, *Numerical optimization.* Springer Science & Business Media, 2006.

[76] S. C. Eisenstat and H. F. Walker, “Choosing the forcing terms in an inexact newton method,” *SIAM J. Sci. Comput.*, vol. 17, no. 1, pp. 16–32, 1996.

[77] S. G. Nash, “A survey of truncated-newton methods,” *J. Comput. Appl. Math.*, vol. 124, no. 1-2, pp. 45–59, 2000.

[78] A. S. Berahas, R. Bollapragada, and J. Nocedal, “An investigation of newton-sketch and subsampled newton methods,” *Optim. Methods. Softw.*, pp. 1–20, 2020.

[79] H. Avron and S. Toledo, “Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix,” *J. ACM*, vol. 58, no. 2, p. 8, 2011.

[80] L. Elden, “Algorithms for the regularization of ill-conditioned least squares problems,” *BIT*, vol. 17, no. 2, pp. 134–145, 1977.

[81] H. D. Simon, “Analysis of the symmetric lanczos algorithm with reorthogonalization methods,” *Linear Algebra Appl.*, vol. 61, pp. 101–131, 1984.

[82] S. Gazzola, P. Novati, and M. R. Russo, “On krylov projection methods and tikhonov regularization,” *Electron. Trans. Numer. Anal*, vol. 44, no. 1, pp. 83–123, 2015.

[83] R. Hunger, “Floating point operations in matrix-vector calculus,” tech. rep., Munich, Germany, 2007.

[84] J. Liu and S. Wright, “An accelerated randomized kaczmarz algorithm,” *Math. Comp.*, vol. 85, no. 297, pp. 153–178, 2016.

[85] Y. Nesterov, “Introductory lectures on convex programming volume i: Basic course,” 1998.

[86] L. Zhang, M. Mahdavi, R. Jin, T. Yang, and S. Zhu, "Recovering the optimal solution by dual random projection," in *Conference on Learning Theory*, pp. 135–157, 2013.

[87] G. H. Golub, F. T. Luk, and M. Overton, "Block lanczos method for computing the singular values and a corresponding singular vectors of a matrix.," *ACM Trans. Math. Software*, vol. 7, no. 2, pp. 149–169, 1981.

[88] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, "Basic linear algebra subprograms for fortran usage," *ACM. Trans. Math. Softw.*, vol. 5, no. 3, pp. 308–323, 1979.

[89] I. K. Ozaslan, M. Pilanci, and O. Arikan, "Fast and robust solution techniques for large scale linear system of equations," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2019.

[90] J. Lacotte and M. Pilanci, "Faster least squares optimization," *arXiv preprint arXiv:1911.02675*, 2019.

[91] S. Lahabar and P. Narayanan, "Singular value decomposition on gpu using cuda," in *2009 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1–10, IEEE, 2009.

[92] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.

[93] L. Elden, "A note on the computation of the generalized cross-validation function for ill-conditioned least squares problems," *BIT*, vol. 24, no. 4, pp. 467–472, 1984.

Appendix A

Discussion on the Proposed Error Upper Bound for Iterations of Primal Dual Algorithms in [1]

In this appendix, we provide details of a critical discussion on the steps of the derivation that leads to an error upper bound for the iterations of the primal dual algorithms given in [1]. First, we provide a short list of minor issues that can easily be corrected.

1. During the initialization stage in *Line 2* of both *Algorithm 4* in page 4097 and *Algorithm 5* in page 4098, the residual error vector $\mathbf{r}^{(0)}$ must be set to $-\lambda\mathbf{y}$ instead of $-\mathbf{y}$, otherwise iterates of the both of the algorithms diverge from the optimal solution.

2. During the initialization stage in *Line 2* of *Algorithm 7* in 4912, the dual residual error vector $\mathbf{r}_{\text{Dual}}^{(0)}$ must be set to $-\lambda\mathbf{y}$ instead of $-\mathbf{y}$ and during the initialization stage of the inner loop iterations in *Line 15*, the primal residual error vector $\mathbf{r}_{\text{P}}^{(0)}$ must be set to $-\mathbf{R}^T\mathbf{X}^T\mathbf{r}_{\text{D}}^{(t+1)}$; otherwise iterates of the algorithm diverges from the optimal solution. The MATLAB codes provided in the link includes these corrections.

In addition to the above mentioned minor issues, there are some major issues as well. Unfortunately, we could not obtain corrective actions on these major issues as we could have done on the minor issues mentioned above. Therefore, a lower bound on the number of inner loop iterations, that guarantee a certain rate of convergence at the main loop, is still an open question for the primal dual algorithms. In the remaining of this appendix, we will provide steps of the derivation presented in [1], along with our critical remarks on their validity.

Consider the following A-IHS updates

$$\widehat{\mathbf{w}}^{t+1} = \widehat{\mathbf{w}}^{t} + \widehat{\mathbf{u}}^{t}.$$

We are going to use exactly the same notation as [1] except for that *HS* subscript for the A-IHS iterates are omitted. In the primal dual algorithms, instead of exact sequence $\{\widehat{\mathbf{w}}^{t}\}$, a sequence $\{\widetilde{\mathbf{w}}^{t}\}$ is obtained due to the approximate minimizers that are used in place of $\widehat{\mathbf{u}}^{t}$. Consequently while the sequence $\{\widehat{\mathbf{w}}^{t}\}$ is obtained after t exact iterations of the A-IHS algorithm, sequence $\{\widetilde{\mathbf{w}}^{t}\}$ is obtained after t primal dual iterations in each of which k inner loop updates are used to approximate $\widehat{\mathbf{u}}^{t}$'s. The details of the inner and outer loops can be found in Algorithm 7 of [1]. The aim of the *Theorem 9* is to establish an upper bound for

$$\left\|\widetilde{\mathbf{w}}^{t+1} - \mathbf{w}^{*}\right\|_{\mathbf{X}}$$

where $\mathbf{w}^{*}$ is the true minimizer of the primal objective function. The triangle inequality and the convergence rate of the A-IHS that is established in *Theorem 2* of [1] is used to find an upper bound for this error norm:

$$\begin{aligned} \left\|\widetilde{\mathbf{w}}^{t+1} - \mathbf{w}^{*}\right\|_{\mathbf{X}} &\leq \left\|\widehat{\mathbf{w}}^{t+1} - \mathbf{w}^{*}\right\|_{\mathbf{X}} + \left\|\widetilde{\mathbf{w}}^{t+1} - \widehat{\mathbf{w}}^{t+1}\right\|_{\mathbf{X}} \\ &\leq \alpha^{t} \left\|\mathbf{w}\right\|_{\mathbf{X}} + \left\|\widetilde{\mathbf{w}}^{t+1} - \widehat{\mathbf{w}}^{t+1}\right\|_{\mathbf{X}}, \end{aligned} \tag{A.1}$$

where $\alpha = \frac{C_0\sqrt{\mathbb{W}^2(\mathbf{X}\mathbb{R}^p \cap \mathcal{S}^{n-1})\log(1/\delta)}}{1 - C_0\sqrt{\mathbb{W}^2(\mathbf{X}\mathbb{R}^p \cap \mathcal{S}^{n-1})\log(1/\delta)}}$. At this point a new iterate, $\overline{\mathbf{w}}^{t+1}$, is introduced, which is the result of one exact step of the IHS initialized at $\widetilde{\mathbf{w}}^{t}$. The inner loop iterations at the t-th outer (main) loop iteration of the primal dual iterations are expected to converge $\overline{\mathbf{w}}^{t+1}$. Therefore,

$$\begin{aligned}
\left\|\widetilde{\mathbf{w}}^{t+1} - \widehat{\mathbf{w}}^{t+1}\right\|_{\mathbf{X}} &\leq \left\|\widetilde{\mathbf{w}}^{t+1} - \overline{\mathbf{w}}^{t+1}\right\|_{\mathbf{X}} + \left\|\overline{\mathbf{w}}^{t+1} - \widehat{\mathbf{w}}^{t+1}\right\|_{\mathbf{X}}, \\
\left\|\widetilde{\mathbf{w}}^{t+1} - \overline{\mathbf{w}}^{t+1}\right\|_{\mathbf{X}} &\leq \lambda_{max}\left(\frac{\mathbf{X}^T\mathbf{X}}{n}\right)\beta^k \left\|\overline{\mathbf{w}}^{t+1}\right\|_2 \\
&\leq \lambda_{max}\left(\frac{\mathbf{X}^T\mathbf{X}}{n}\right)\beta^k \left(\left\|\overline{\mathbf{w}}^{t+1} - \mathbf{w}^*\right\|_2 + \left\|\mathbf{w}^*\right\|_2\right) \\
&\leq 2\lambda_{max}\left(\frac{\mathbf{X}^T\mathbf{X}}{n}\right)\beta^k \left\|\mathbf{w}^*\right\|_2, \qquad \text{(A.2)}
\end{aligned}$$

where $\beta = \frac{C_0\sqrt{\mathbb{W}^2(\mathbf{X}^T\mathbb{R}^p\cap\mathcal{S}^{p-1}})\log(1/\delta)}{1-C_0\sqrt{\mathbb{W}^2(\mathbf{X}\mathbb{R}^p\cap\mathcal{S}^{p-1}})\log(1/\delta)}$. The last inequality is not valid unless

$$\left\|\overline{\mathbf{w}}^{t+1} - \mathbf{w}^*\right\|_2 \leq \left\|\mathbf{w}^*\right\|_2.$$

However, particularly during the initial phases of the main iterations this condition can be violated. Therefore, this step of the proof requires a major revision. Assuming that such revision is possible, up to this point, the following is obtained:

$$\left\|\widetilde{\mathbf{w}}^{t+1} - \mathbf{w}^*\right\|_{\mathbf{X}} \leq \alpha^t \left\|\mathbf{w}\right\|_{\mathbf{X}} + 2\lambda_{max}\left(\frac{\mathbf{X}^T\mathbf{X}}{n}\right)\beta^k \left\|\mathbf{w}^*\right\|_2 + \left\|\overline{\mathbf{w}}^{t+1} - \widehat{\mathbf{w}}^{t+1}\right\|_{\mathbf{X}}. \qquad \text{(A.3)}$$

To proceed for the final form of the upper bound, the following upper bound on the last term of eq. (A.3) is given in [1]:

$$\left\|\overline{\mathbf{w}}^{t+1} - \widehat{\mathbf{w}}^{t+1}\right\|_{\mathbf{X}} \leq \left\|\widetilde{\mathbf{H}}^{-1}\right\|_2 \left\|\widetilde{\mathbf{H}} - \mathbf{H}\right\|_2 \left\|\widetilde{\mathbf{w}}^t - \widehat{\mathbf{w}}^t\right\|_{\mathbf{X}} \leq \frac{4\lambda_{max}\left(\frac{\mathbf{X}^T\mathbf{X}}{n}\right)}{\lambda} \left\|\widetilde{\mathbf{w}}^t - \widehat{\mathbf{w}}^t\right\|_{\mathbf{X}},$$

which is a valid bound. Then in [1] the following upper bound is given without necessary justification:

$$\left\|\widetilde{\mathbf{w}}^t - \widehat{\mathbf{w}}^t\right\|_{\mathbf{X}} \leq 2\lambda_{max}\left(\frac{\mathbf{X}^T\mathbf{X}}{n}\right)\beta^k \left\|\mathbf{w}^*\right\|_2.$$

to reach the final form of the error upper bound:

$$\left\|\widetilde{\mathbf{w}}^{t+1} - \mathbf{w}^*\right\|_{\mathbf{X}} \leq \alpha^t \left\|\mathbf{w}^*\right\|_{\mathbf{X}} + \frac{10\lambda_{max}^2\left(\frac{\mathbf{X}^T\mathbf{X}}{n}\right)}{\lambda}\beta^k \left\|\mathbf{w}^*\right\|_2.$$

However, this final form of the upper bound is not supported in detail as part of the presented proof. Because of the following major issue, we conclude that the proposed bound remains an unproven conjecture. The bound established for $\|\widetilde{\mathbf{w}}^{t+1} - \overline{\mathbf{w}}^{t+1}\|_{\mathbf{X}}$ in eq. (A.2) is used to upper bound $\|\widetilde{\mathbf{w}}^{t} - \widehat{\mathbf{w}}^{t}\|_{\mathbf{X}}$. This is not justified as part of the proof in [1].

Appendix B

TSVD Based Regularization Scheme

Following theorem provides an example algorithm that inherits the convergence behaviour given in eq. (4.14):

Theorem B.0.1. *Consider the TSVD solution with truncation parameter k:*

$$\mathbf{x}(k) = \mathbf{V}_k \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T b$$

The modified `Hybrid M-IHS` *updates,*

$$\begin{aligned} \Delta\mathbf{x}^i(k_i) &= \mathbf{V}_{k_i} \mathbf{\Sigma}_{k_i}^{-1} \left(\mathbf{U}_{k_i}^T \mathbf{S}^T \mathbf{S} \mathbf{U}_{k_i}\right)^{-1} \left(\mathbf{U}_{k_i}^T \mathbf{b} - \mathbf{\Sigma}_{k_i} \mathbf{V}_{k_i}^T \mathbf{x}^i\right) \\ \mathbf{x}^{i+1} &= \mathbf{x}^i + \alpha_i \Delta\mathbf{x}^i(k_i) + \beta_i(\mathbf{x}^i - \mathbf{x}^{i-1}) \end{aligned}$$

with truncation parameters $k_0 \leq \ldots \leq k_i = k$ and momentum parameters

$$\beta_i = k_i/m, \qquad \alpha_i = (1-\beta_i)^2,$$

converge to the TSVD solution $\mathbf{x}(k)$ *at the following rate:*

$$\|\mathbf{x}^{i+1} - \mathbf{x}(k)\|_2 \leq \sum_{\ell=0}^{i} \left(\prod_{j=\ell}^{i} \sqrt{\beta_j} \right) \frac{\sigma_{k_{\ell-1}+1}}{\sigma_{k_\ell}} \left\|\mathbf{x}(k_{\ell-1} : k_\ell)\right\|_2 ,$$

where $k_{-1} = 0$ *and the initial guess* $\mathbf{x}^0 = \mathbf{0}$.

Proof. Firstly assume that constant truncation parameter k and momentum parameters $\beta = k/m$, $\alpha = (1-\beta)^2$ are used for all iterations. Consider the following bipartite transformation

$$\begin{bmatrix} \mathbf{\Sigma}_k \mathbf{V}_k \left(\mathbf{x}^{i+1} - \mathbf{x}(k)\right) \\ \mathbf{\Sigma}_k \mathbf{V}_k \left(\mathbf{x}^{i} - \mathbf{x}(k)\right) \end{bmatrix} = \underbrace{\begin{bmatrix} (1+\beta)\mathbf{I}_k - \alpha \left(\mathbf{U}_k^T \mathbf{S}^T \mathbf{S} \mathbf{U}_k\right)^{-1} & \beta \mathbf{I}_k \\ \mathbf{I}_d & 0 \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} \mathbf{\Sigma}_k \mathbf{V}_k \left(\mathbf{x}^{i} - \mathbf{x}(k)\right) \\ \mathbf{\Sigma}_k \mathbf{V}_k \left(\mathbf{x}^{i-1} - \mathbf{x}(k)\right) \end{bmatrix}.$$

If the inequality

$$\beta \geq \left(1 - \sqrt{\alpha \psi_i}\right)^2, \forall i \in [r] \tag{B.1}$$

holds, then all eigenvalues of the bipartite transformation are imaginary and have magnitude $\sqrt{\beta}$, where ψ_i is the i^{th} eigenvalue of $\left(\mathbf{U}_k^T \mathbf{S}^T \mathbf{S} \mathbf{U}_k\right)^{-1}$. The ψ_i values can be stochastically bounded by using the Approximate Matrix Property of the sketch matrices:

$$\mathbb{P}_{\mathbf{S} \sim \mathcal{D}} \left(\left\| \mathbf{U}_k^T \mathbf{S}^T \mathbf{S} \mathbf{U}_k - \mathbf{I}_k \right\|_2 \leq \epsilon \right) \leq \delta. \tag{B.2}$$

where the parameters ϵ and δ can be chosen according to Lemma 3.2.2. We refer interested reader to Section 3.2 for further discussion about the property and we move on with the asymptotic bounds for simplicity. Since $\mathbf{U}_k$ is an orthogonal transformation, according to the MPL, the largest and smallest eigenvalues of $\mathbf{U}_k^T \mathbf{S}^T \mathbf{S} \mathbf{U}_k$ is asymptotically bounded in the interval $[(1-\sqrt{k/m})^2,\ (1+\sqrt{k/m})^2]$ as $m \to \infty$ while the ratio k/m remains constant [63]. Consequently, the condition in eq. (B.1) can be satisfied for all ψ_i's by the following choice of β that maximizes the convergence rate over step size α

$$\sqrt{\beta^*} = \underset{\alpha}{\text{minimize}} \ \max \left\{ \left| 1 - \frac{\sqrt{\alpha}}{1 + \sqrt{k/m}} \right|, \left| 1 - \frac{\sqrt{\alpha}}{1 - \sqrt{k/m}} \right| \right\} = \sqrt{\frac{k}{m}}$$

where the minimum is achieved at $\alpha^* = (1 - \frac{k}{m})^2$. As a result, we have the following inequality:

$$\|\mathbf{\Sigma}_k \mathbf{V}_k^T \left(\mathbf{x}(k) - \mathbf{x}^{i+1}\right)\|_2 \leq \sqrt{\frac{k}{m}} \left\|\mathbf{\Sigma}_k \mathbf{V}_k^T \left(\mathbf{x}(k) - \mathbf{x}^i\right)\right\|_2$$

which satisfies the condition in theorem 4.2.1. If the parameters are changing through the iterations, then consider the transformed error at the $(i+1)^{th}$ iteration:

$$\begin{aligned}
\mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}^{i+1} - \mathbf{x}(k)) &= \mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}^i - \mathbf{x}(k)) - \alpha_i \left(\mathbf{U}_{k_i}^T\mathbf{S}^T\mathbf{S}\mathbf{U}_{k_i}\right)^{-1} \mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}^i - \mathbf{x}(k)) \\
&\quad + \beta_i \mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}^i - \mathbf{x}^{i-1}) \\
&\overset{(i)}{=} \mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}^i - \mathbf{x}(k_{i-1})) \\
&\quad - \alpha_i \left(\mathbf{U}_{k_i}^T\mathbf{S}^T\mathbf{S}\mathbf{U}_{k_i}\right)^{-1} \mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}^i - \mathbf{x}(k_{i-1})) \\
&\quad + \beta_i \mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}^i - \mathbf{x}^{i-1}) \\
&\quad \overset{(ii)}{+} \mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}(k_{i-1} : k_i)) \\
&\quad + \alpha_i \left(\mathbf{U}_{k_i}^T\mathbf{S}^T\mathbf{S}\mathbf{U}_{k_i}\right)^{-1} \mathbf{\Sigma}_{k_i}\mathbf{V}_{k_i}^T(\mathbf{x}(k_{i-1} : k_i))
\end{aligned}$$

where $\mathbf{x}(k_{i-1} : k_i) = \mathbf{x}(k_i) - \mathbf{x}(k_{i-1})$. The above Lyapunov analysis can be applied to the linear transformation indicated by (i) and (ii) independently since two error components

$$\left(\mathbf{x}^i - \mathbf{x}(k_{i-1})\right) \in \mathbf{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{k_{i-1}}\} \text{ and } \mathbf{x}(k_{i-1} : k_i) \in \mathbf{span}\{\mathbf{v}_{k_{i-1}+1}, \ldots, \mathbf{v}_{k_i}\}$$

are orthogonal to each other, which gives

$$\begin{aligned}
\left\|\mathbf{V}_{k_{i-1}}^T(\mathbf{x}^{i+1} - \mathbf{x}(k_{i-1}))\right\|_2 &\leq \beta_i \frac{\sigma_1}{\sigma_{k_{i-1}}} \left\|\mathbf{V}_{k_{i-1}}^T(\mathbf{x}^i - \mathbf{x}(k_{i-1}))\right\|_2, \\
\left\|\mathbf{V}_{k_{i-1}+1:k_i}^T(\mathbf{x}^{i+1} - \mathbf{x}(k_{i-1} : k_i))\right\|_2 &\leq \beta_i \frac{\sigma_{k_i}}{\sigma_{k_{i-1}+1}} \left\|\mathbf{V}_{k_{i-1}+1:k_i}^T \mathbf{x}(k_{i-1} : k_i))\right\|_2,
\end{aligned}$$

where we used the fact that $\mathbf{x}^{i+1} \in \text{span}(\mathbf{V}_{k_i})$ for $0 \leq i \leq r$ if initial guess $\mathbf{x}^0 = \mathbf{0}$. Applying the above splitting to all iterations recursively results in the proposed upper bound of eq. (4.14). □

Appendix C

Bidiagonalization Based Implementations of the Hybrid M-IHS Techniques

For the derivations of the proposed hybrid methods, we used the SVD of the sketched matrices, but the SVD may not be preferred in practical applications due to relatively high computational complexity. In this subsection, we provide the GKL bidiagonalization-based versions of the proposed techniques given in Chapter 4 that require only matrix-vector and vector-vector multiplications, i.e., level 1 and 2 of BLAS operations.

For all proposed techniques, we use the *bidiag2* procedure, which is described in eq. (3.25). The input matrix $\mathbf{A}$ in eq. (3.25) will be replaced by one of the sketched $\mathbf{SA}$, $\mathbf{SA}^T$ and $\mathbf{WAS}^T$ matrices according to the variant of the interest. The upper bidiagonal decomposition can be computed by using the three term recurrence given in eq. (3.26) by substituting an appropriate matrix for $\mathbf{A}$ and setting the starting vector $\theta_1\mathbf{q}_1$ to one of the vectors $(\mathbf{SA})^T\mathbf{Sb}$, $\mathbf{A}^T\mathbf{b}$ and $\mathbf{SA}^T\mathbf{b}$, respectively, for the `Hybrid M-IHS`, the `Hybrid Dual M-IHS` and the `Hybrid Primal Dual M-IHS`. To preserve the norm of the RHS vectors of the sub-problems, reorthogonalization must be applied at least to $\mathbf{Q}_k$ matrices. The classic or the modified Gram Schmidt processes can be used for this purpose [50]. Noting that the matrix $\mathbf{P}_k$ and $\mathbf{R}_k$ are not directly needed for any of the proposed methods, so they may not be assembled at all. The `Hybrid M-IHS` and the `Hybrid Primal Dual M-IHS` use the triangular matrix $\mathbf{T}_k := \mathbf{R}_k\mathbf{R}_k^T$ instead of the bidiagonal form $\mathbf{R}_k$ and the matrix $\mathbf{D}_k := \mathbf{V}_k\mathbf{R}_k^{-1}$ which can be constructed during the iterations of the GKL procedure as

$$\mathbf{d}_j = \left(\mathbf{q}_j - \theta_j\mathbf{d}_{j-1}\right)/\rho_j, \text{ where } j \in [k] \text{ and } \mathbf{d}_0 = \mathbf{0}.$$

The `Hybrid Dual M-IHS`, on the other side, needs only the triangular form $\widetilde{\mathbf{T}}_k = \mathbf{R}_k^T\mathbf{R}_k$ and the orthonormal transformation $\mathbf{Q}_k$. Once the proper pair of matrices are constructed at the very beginning of iterations, any other GKL-related matrices or sketched matrices $\mathbf{SA}, \mathbf{AS}^T, \mathbf{WAS}^T$ will not be needed anymore and can be removed from the memory.

The risk estimator given in eq. (4.10) at the i-th iteration of the `Hybrid M-IHS` can be computed as:

$$\mathbb{V}_{1,k}(\lambda) = \frac{\left\|\left(\mathbf{T}_k + \lambda\mathbf{I}_k\right)^{-1}\mathbf{f}_i\right\|_2}{\mathsf{tr}\left(\left(\mathbf{T}_k + \lambda\mathbf{I}\right)^{-1}\right)}, \text{ where } \mathbf{f}_i = \mathbf{D}_k^T\mathbf{g}^i + \mathbf{T}_k\mathbf{D}_k^T\mathbf{x}^i \tag{C.1}$$

where the regularized HS step is:

$$\Delta\mathbf{x}^i(\lambda_i) = \mathbf{D}_k\mathbf{T}\left(\mathbf{T}_k + \lambda_i\mathbf{I}_k\right)^{-1}\mathbf{D}_k^T\left(\mathbf{g}^i - \lambda_i\mathbf{x}^i\right).$$

The risk estimator in eq. (4.20) at the i-th iteration of the `Hybrid Dual M-IHS` can be computed as:

$$\mathbb{V}_{2,k}(\lambda) = \frac{\left\|\left(\widetilde{\mathbf{T}}_k + \lambda\mathbf{I}\right)^{-1}\mathbf{f}_i\right\|_2}{\mathsf{tr}\left(\left(\widetilde{\mathbf{T}}_k + \lambda\mathbf{I}_k\right)^{-1}\right)}, \text{ where } \mathbf{f}_i = \mathbf{Q}_k^T\mathbf{h}^i + \widetilde{\mathbf{T}}_k\mathbf{Q}_k^T\boldsymbol{\nu}^i \tag{C.2}$$

where the regularized HS step is

$$\Delta\boldsymbol{\nu}^i(\lambda_i) = \mathbf{Q}_k\left(\widetilde{\mathbf{T}}_k + \lambda_i\mathbf{I}_k\right)^{-1}\mathbf{Q}_k^T\left(\mathbf{h}_i - \lambda_i\boldsymbol{\nu}^i\right).$$

The risk estimator in eq. (4.30) at the j-th inner loop iteration of the i-th outer loop of the `Hybrid Primal Dual M-IHS` can be computed as

$$\mathbb{V}_{3,k}(\lambda) = \frac{\left\|\left(\mathbf{T}_k + \lambda \mathbf{I}\right)^{-1} \mathbf{f}_{i,j}\right\|_2}{\mathsf{tr}\left(\left(\mathbf{T}_k + \lambda \mathbf{I}_k\right)\right)} \text{ where } \mathbf{f}_{i,j} = \mathbf{D}_k^T \mathbf{g}^{i,j} + \mathbf{T}_k \mathbf{D}_k^T \left(\mathbf{z}^{i,j} + \mathbf{SA}^T \boldsymbol{\nu}^i\right) \quad \text{(C.3)}$$

where the regularized HS step is

$$\Delta \mathbf{z}^{i,j}(\lambda_{i,j}) = \mathbf{D}_k \mathbf{T}_k \left(\mathbf{T}_k + \lambda_{i,j} \mathbf{I}_k\right)^{-1} \mathbf{D}_k^T \left(\mathbf{g}^{i,j} - \lambda_{i,j}\left(\mathbf{z}^{i,j} + \mathbf{SA}^T \boldsymbol{\nu}^i\right)\right).$$

The GKL based algorithms are given in Algorithm 12, 13 and 14, respectively.

Algorithm 11 Finding trace of the inverse symmetric tridiagonal matrix

```
*Input:* $\mathbf{T} \in \mathbb{R}^{L \times L}$ *is a symmetric tridiagonal matrix*
$\rho = \mathbf{T}_{L,L}, \quad \tau_L = 1/\rho$
**for** $i = L-1 : -1 : 1$ **do**
    $\theta = \mathbf{T}_{i+1,i}/\rho, \quad \rho = \mathbf{T}_{i,i} - \theta$
    $\tau_i = (1 + \theta \cdot \tau_{i+1})/\rho$
**end for**
*Output:* $\mathsf{tr}\left(\mathbf{T}^{-1}\right) = \sum_i^L \tau_i$
```

Algorithm 12 `Hybrid M-IHS` (for $n \gg d$)

```
*Input:* $\mathbf{A} \in \mathbb{R}^{n \times d}, \ \mathbf{b}, \ m, \ \mathbf{x}^0$
$[\mathbf{SA}, \ \mathbf{Sb}] = \texttt{RP\_fun}(\mathbf{A}, \ \mathbf{b}, \ m)$
    $[\mathbf{T}, \mathbf{D}] = \texttt{GKL\_fun}(\mathbf{SA}, \ \mathbf{AS}^T\mathbf{Sb}, \ d)$
**while** *until stopping criteria* **do**
    $\widetilde{\mathbf{g}}^i = \mathbf{D}^T \mathbf{A}^T \left(\mathbf{b} - \mathbf{A}\mathbf{x}^i\right)$
    $\widetilde{\mathbf{x}}^i = \mathbf{D}^T \mathbf{x}^i$
    $\mathbf{f}^i = \widetilde{\mathbf{g}}^i + \mathbf{T}^T \widetilde{\mathbf{x}}^i$
    $\lambda_i = \underset{\lambda}{\operatorname{argmin}} \ \mathbb{V}_1(\lambda)$ *given in* eq. (C.1)
    $\Delta \mathbf{x}^i = \mathbf{DT}\left(\mathbf{T} + \lambda_i \mathbf{I}_d\right)^{-1} \left(\widetilde{\mathbf{g}}^i - \lambda_i \widetilde{\mathbf{x}}^i\right)$
    $\widehat{k} = d - \lambda_i \cdot \mathsf{tr}\left(\left(\mathbf{T} + \lambda_i \mathbf{I}\right)^{-1}\right)$
    $\beta_i = \widehat{k}/m, \quad \alpha_i = (1 - \beta_i)^2$
    $\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha_i \Delta \mathbf{x}^i + \beta_i(\mathbf{x}^i - \mathbf{x}^{i-1})$
**end while**
```

Algorithm 13 `Hybrid Dual M-IHS` (for $n \ll d$)

```
Input: $\mathbf{A} \in \mathbb{R}^{n \times d},\ \mathbf{b},\ m$
  $\mathbf{SA}^T = \texttt{RP\_fun}(\mathbf{A}^T, m)$
$[\overline{\mathbf{T}},\ \mathbf{Q}] = \texttt{GKL\_fun}(\mathbf{SA}^T,\ \mathbf{b},\ n)$
    $\boldsymbol{\nu}^0 = \mathbf{x}^0 = \mathbf{0}$
while until stopping criteria do
    $\widetilde{\mathbf{h}}^i = \mathbf{Q}^T\left(\mathbf{b} - \mathbf{A}\mathbf{x}^i\right)$
    $\widetilde{\boldsymbol{\nu}}^i = \mathbf{Q}^T\widetilde{\boldsymbol{\nu}}^i$
  $\mathbf{f}^i = \widetilde{\mathbf{h}}^i + \overline{\mathbf{T}}\widetilde{\boldsymbol{\nu}}^i$
    $\lambda_i = \underset{\lambda}{\operatorname{argmin}}\ \mathbb{V}_2(\lambda)$ given in eq. (C.2)
  $\Delta\boldsymbol{\nu}^i = \mathbf{Q}\left(\overline{\mathbf{T}} + \lambda_i\mathbf{I}_d\right)^{-1}\left(\widetilde{\mathbf{h}}^i - \lambda_i\widetilde{\boldsymbol{\nu}}^i\right)$
    $\widehat{k} = d - \lambda_i \cdot \mathsf{tr}\left(\left(\overline{\mathbf{T}} + \lambda_i\mathbf{I}\right)^{-1}\right)$
    $\beta_i = \widehat{k}/m, \quad \alpha_i = (1-\beta_i)^2$
  $\boldsymbol{\nu}^{i+1} = \boldsymbol{\nu}^i + \alpha_i\Delta\boldsymbol{\nu}^i + \beta_i(\boldsymbol{\nu}^i - \boldsymbol{\nu}^{i-1})$
  $\mathbf{x}^{i+1} = \mathbf{A}^T\boldsymbol{\nu}^{i+1}$
end while
```

The GKL procedure which is applied only once at the beginning of the algorithms can be computed in $O(m\min(n,d)^2)$ (or $O(m_2 m_1^2)$) operations for dense matrices. The inversions at the numerator of proposed estimators given in eq. (C.1), eq. (C.2) and eq. (C.3) can be computed in $O(k)$ operations by the LU factorization or the Givens Rotations thanks to the tridiagonal form. The trace terms in the denominators can be calculated in **6k** operations by modifying the technique suggested by Elden in [93] as demonstrated in Algorithm 11.

As a result, for a given λ value, the proposed risk estimators can be calculated in $O(k)$ operations. Besides, most of the terms in the regularized HS steps are already required for the risk estimators, hence the HS steps require only one tridiagonal matrix-vector multiplication and one dense matrix-vector multiplication in addition to solving one triangular system. The MATLAB implementation of the algorithms described in this section are also provided in the following link: `https://github.com/ibrahimkurban/Hybrid-M-IHS`.

Algorithm 14 `Hybrid Primal Dual M-IHS` (for $n \leq d$ or $n \geq d$)

```
Input: $\mathbf{A} \in \mathbb{R}^{n \times d},\ \mathbf{b},\ m_1,\ m_2$
$[\mathbf{S}\mathbf{A}^T] = \texttt{RP\_fun}(\mathbf{A}^T,\ m_1)$
$[\mathbf{W}\mathbf{A}\mathbf{S}^T] = \texttt{RP\_fun}(\mathbf{A}\mathbf{S}^T,\ m_2)$
$[\mathbf{T}, \mathbf{D}] = \texttt{GKL\_fun}(\mathbf{W}\mathbf{A}\mathbf{S}^T,\ \mathbf{S}\mathbf{A}^T\mathbf{b},\ m_1)$
$\tau = -\infty$, $i = -1$, $\boldsymbol{\nu}^0 = \mathbf{x}^0 = \mathbf{0}$, $\mathbf{z}^{0,0} = \mathbf{0}$
while until first stopping criteria do
    $i = i + 1$
    $\mathbf{h}^i = \mathbf{b} - \mathbf{A}\mathbf{x}^i$
    $\widetilde{\boldsymbol{\nu}}^i = \mathbf{S}\mathbf{A}^T\boldsymbol{\nu}^i$    (or $\widetilde{\boldsymbol{\nu}}^i = \mathbf{S}\mathbf{x}^i$)
    $\mathbf{z}^{i,0} = \mathbf{z}^{i-1,j}$, $j = -1$
    while until second stopping criteria do
        $j = j + 1$;
        $\widetilde{\mathbf{g}}^{i,j} = \mathbf{D}^T\mathbf{S}\mathbf{A}^T(\mathbf{h}^i - \mathbf{A}\mathbf{S}^T\mathbf{z}^{i,j})$
        $\widetilde{\mathbf{z}}^{i,j} = \mathbf{D}^T(\mathbf{z} + \widetilde{\boldsymbol{\nu}}^i)$
        $\mathbf{f}^{i,j} = \widetilde{\mathbf{g}}^{i,j} + \mathbf{T}^T\widetilde{\mathbf{z}}^{i,j}$
        $\lambda_{i,j} = \underset{\lambda \in [\tau,\ \infty]}{\operatorname{argmin}}\ \mathbb{V}_3(\lambda)$ given in eq. (C.3)
        $\Delta\mathbf{z}^{i,j} = \mathbf{D}\mathbf{T}\left(\mathbf{T} + \lambda_i\mathbf{I}_d\right)^{-1}\left(\widetilde{\mathbf{g}}^{i,j} - \lambda_{i,j}\widetilde{\mathbf{z}}^{i,j}\right)$
        $\widehat{k} = m_1 - \lambda_{i,j} \cdot \mathsf{tr}\left((\mathbf{T} + \lambda_{i,j}\mathbf{I})^{-1}\right)$
        $\beta_{1,j} = \widehat{k}/m_2, \quad \alpha_{1,j} = (1 - \beta_{1,j})^2$
        $\mathbf{z}^{i,j+1} = \mathbf{z}^{i,j} + \alpha_{1,j}\Delta\mathbf{z}^{i,j} + \beta_{1,j}(\mathbf{z}^{i,j} - \mathbf{z}^{i,j-1})$
    end while
    $\Delta\boldsymbol{\nu}^i = (\mathbf{h}^i - \lambda_{i,j}\boldsymbol{\nu}^i - \mathbf{A}\mathbf{S}^T\mathbf{z}^{i,j})/\lambda_{i,j}$
    $\beta_{2,i} = \widehat{k}/m_1, \quad \alpha_{2,i} = (1 - \beta_{2,i})^2$
    $\boldsymbol{\nu}^{i+1} = \boldsymbol{\nu}^i + \alpha_{2,i}\Delta\boldsymbol{\nu}^i + \beta_{2,i}(\boldsymbol{\nu}^i - \boldsymbol{\nu}^{i-1})$
    $\mathbf{x}^{i+1} = \mathbf{A}^T\boldsymbol{\nu}^{i+1}$
    $\tau = \max(\lambda_{i,j},\ \tau)$
end while
```