Distribution Agreement

Signature:

Xiaoyun Gong April 3rd, 2023

Low Precision Preconditioning for Iterated Tikhonov Regularization

By

Xiaoyun Gong

James Nagy Ph.D.
Advisor

Mathematics

James Nagy Ph.D.
Advisor

Jinho Choi, Ph.D.
Committee Member

Yuanzhe Xi, Ph.D.
Committee Member

2023

Low Precision Preconditioning for Iterated Tikhonov Regularization

By

Xiaoyun Gong

James Nagy Ph.D.
Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences of
Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Mathematics

2023

Abstract

Low Precision Preconditioning for Iterated Tikhonov Regularization
By Xiaoyun Gong

Mixed precision arithmetic has gained significant interest in recent years, given its ability to reduce memory cost and accelerate computation while maintaining accuracy. Many mixed precision algorithms have been designed for solving large-scale, well-conditioned linear systems that arise in various scientific applications. Iterative refinement is a common scheme in the design of such algorithms. In this thesis, we aim to extend mixed precision to ill-conditioned problems using variations of iterated Tikhonov as regularization. Several numerical experiments are conducted on applications from signal and image processing, and the results are compared with those obtained from standard methods, such as CGLS and Hybrid LSQR. Analysis of the results show that the method is able to produce solutions of comparable quality to the standard methods, but at a significantly lower computational cost.

Low Precision Preconditioning for Iterated Tikhonov Regularization

By

Xiaoyun Gong

James Nagy Ph.D.
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Mathematics

2023

Acknowledgments

I would like to thank my advisor Dr. James Nagy for all his guidance and support during my journey at Emory. I am so grateful for everything he has done to help me grow both academically and as a person, for leading me to explore my interest for mathematics and encouraging and supporting me during my times of confusion. I also want to thank my committee members, Dr. Yuanzhe Xi and Dr. Jinho Choi, who also guided me towards discovering my interest and provided me with a lot of support. Their passions are always so inspiring and I learned so much from them.

I feel so fortunate having met such wonderful professors at Emory, and I am thankful to many other professors in and outside the department. I also want to thank my friends who have always been there for me, and with whom I have shared many happy moments. These will be experiences I will never forget.

I would like to thank my parents and family who always support my decisions, and offer me advice. Their love and encouragement are my constant source of energy. I know I can always turn to them. Thank you for always believing in me and being there for me.

Lastly, I want to thank the lovely sunshine of Georgia, the beautiful clouds at sunset, woods in autumn, cats at Cox Bridge, Northern Cardinal sometimes, and squirrels anywhere.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Inverse problems arise in many areas of science and engineering, where people hope to recover unknown information from indirect measurements. These problems often involve solving large-scale linear systems, which can be computationally expensive and prone to accumulation of errors, especially when dealing with noisy data. Using an iterative method is often a popular way for solving such problems. On the other hand, low precision arithmetic has gained significant interest in recent years due to advances in computer architecture and growing demands in scientific computation, as well as hardware support. It has been shown to be particularly useful in solving large-scale, well-conditioned linear systems that arise in various application fields, including engineering, physics, and data science. This is because low precision arithmetic reduces the memory cost and accelerates computation through reducing the number of bits used to represent the numbers in the computer. Therefore people have been exploring the potential of combining low precision arithmetic with iterative methods to tackle those increasingly large scale computational problems. Mixed precision algorithms have been developed that leverage the benefits of low precision arithmetic, such as faster computation and lower memory cost, while still maintaining the accuracy of high precision arithmetic. A comprehensive review of mixed precision algorithms in numerical linear algebra can be found in [16]. Many works have been done that merge this idea with iterative refinement, which will be discussed in more details in Chapter 2. For example, in [5], Carson and Higham proposed a three-precision algorithm for solving nonsingular linear systems with iterative refinement. Through efficient half precision implementation, they demonstrated that the algorithm could achieve both faster computation and even improved accuracy.

Although low precision arithmetic has been studied extensively in well-conditioned linear systems, to our knowledge little work has been done on its application to ill-posed problems. Iterative refinement has been shown to be effective for well-conditioned linear systems, but for solving ill-conditioned problems, regularization is necessary to balance signal and noise. Common techniques for regularization include truncated singular value decomposition (TSVD) and Tikhonov regularization. Therefore, we aim to extend the work on iterative refinement to iterated Tikhonov regularization as a natural starting point for developing mixed-precision algorithms for ill-posed problems. More specifically, our focus is on the iterated Tikhonov scheme and its variations, including Donatelli and Hanke's scheme proposed in [8]. Our approach is mainly inspired by mixed precision in iterative refinement, which involves treating a low precision matrix as a computationally efficient approximation that is sufficiently close to the original one, thus improving computational efficiency while maintaining accuracy.

In Chapter 2, we give a brief overview of inverse problems and regularization techniques, along with low precision arithmetic. In Chapter 3, we delve deeper into one particular type of regularization, iterated Tikhonov, and review some of its variations. Additionally, we discuss the motivation behind our approach and derive the value of the paramter for a spectral equivalence condition, which is used in numerical experiments in Chapter 4. The experiments include a spectra signal deconvolution problem and an image deblurring problem, and outcomes are analyzed and summarized in Chapter 5.

Chapter 2

Background

In this chapter we provide some necessary background before we move on to later chapters. Specifically, this chapter covers important topics such as inverse problems, regularization, and low-precision arithmetic.

2.1 Inverse Problems and Regularization

Inverse problems refer to problems that use outside measurements to acquire information about internal or hidden data [14]. Two problems are inverses of each other if the formulation of one problem involves the other one [18]. Distinctions of "forward problem" versus "backward problems" are sometimes blurred. For historical reasons, one of the two problems is better studied, making the less-studied one the inverse/backward problem. As illustrated in Figure 2.1, the forward problem inputs x through the model to obtain b. While for the backward problem, there are two possibilities: (1) Given b and the model, we want to recover the input x; (2) Given input x and output b, we hope to reconstruct the model in between. In real world problems there is often a natural distinction between the two [15]. Inverse problems arise in many application fields, such as geophysical sciences, medical imaging, and many other aspects of engineering. Seismic inversion is one such example where the objective is to deduce the physical characteristics of the subsurface by analyzing seismic measurements. In this case, output measurements and the model, which is determined by the measuring tools, are given, and the goal is to estimate the input.

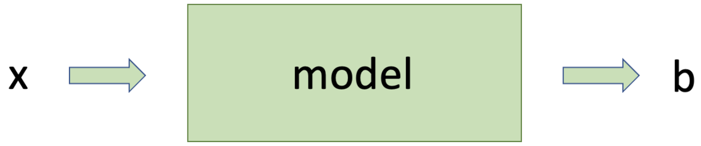

Figure 2.1: forward and backward problems

Inverse problems can be linear or non-linear depending on the formulation and assumptions of the actual problem. In this thesis we only consider linear inverse problems, which occur frequently in applications of tomography [2], radar [26], sonar [23], optical imaging [11], particle sizing [22] and so on. Such problems often involve solving a linear system with noise on the observed data

$$Ax = b = b^{exact} + e,$$

where A is a large-scale, typically ill-conditioned matrix and b is a vector output blended with noise e. In this case we are solving a backward problem of type (1), and the matrix A here is the model.

Direct methods such as LU factorization, Choleskly factorization and QR factorization can be used to calculate a naive solution to the problem $Ax = b$. However the naive solution obtained is often corrupted by noise due to the ill-conditioning of matrix A. Consequently, regularization methods are often needed to balance signal and noise. Yet still, for large scale problems, the cost for factorization is computationally expensive and takes up too much storage. Direct methods also fail to take advantage of the sparsity pattern of the matrix as most direct factorizations do not preserve sparsity, which further increases storage burden.

An iterative method is naturally a better choice in this case, as it only requires matrix-vector products and vector operations. For sparse matrices that have a lot of zero entries, matrix-vector multiplication would be very efficient and therefore speeds up the process. Fast matrix-vector multiplication can also be done for certain structured matrices, such as circulant matrices, Toeplitz matrices, and matrices that can be decomposed into a sum of Kronecker products. We can also incorporate regularization into the iterative method to lead to more stable approximate solutions [14].

Below we describe several commonly-used regularization methods.

2.1.1 Truncated SVD

The Truncated Singular Value Decomposition (TSVD) approximates the matrix A with a close lower rank version of itself that replaces small singular values with zeros to reduce the effect of noise on the final solution.

Consider the SVD of matrix A:

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

where U and V are orthogonal matrices whose columns are, respectively, the left and right singular vectors of A, and the diagonal elements of Σ are the singular values of A ordered from largest to smallest as follows:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0,$$

where r is the rank of A.

If we take a direct approach to get the solution in the case A is a nonsingular matrix, we have:

$$\begin{aligned} A^{-1}b &= A^{-1}b^{exact} + A^{-1}e \\ &= V\Sigma^{-1}U^T b^{exact} + V\Sigma^{-1}U^T e \\ &= x^{exact} + \sum_{i=1}^{n} \frac{u_i^T e}{\sigma_i} v_i. \end{aligned}$$

Notice for small σ_i, the term $\frac{u_i^T e}{\sigma_i} v_i$ in the error term could be large. Therefore the idea of TSVD is to truncate part of SVD of A that has small singular values. The original problem is replaced by inverting the first k singular components of A, and the solution becomes:

$$x_k = \sum_{i=1}^{k} \frac{u_i^T b^{exact}}{\sigma_i} v_i + \sum_{i=1}^{k} \frac{u_i^T e}{\sigma_i} v_i.$$

We can also write it as

$$x_k = \sum_{i=1}^{n} \phi_i^{[k]} \frac{u_i^T b}{\sigma_i} v_i$$

where the filter factors $\phi_i^{[k]}$ are

$$\phi_i^{[k]} = \begin{cases} 0, & i > k \\ 1, & i \leq k \end{cases}.$$

In this way, we control the effect of noise on the solution at the expense of sacrificing part of the information about the real solution. The bias we introduced by TSVD is:

$$\sum_{i=k+1}^{n} \frac{u_i^T b^{exact}}{\sigma_i} v_i = \sum_{i=k+1}^{n} v_i^T x^{exact} v_i$$

When the discrete Picard condition is satisfied, $|v_i^T x^{exact}|$ are typically small compared to x^{exact} [14].

The truncation parameter k is based on how much we weigh noise against bias. The choice of k is determined by the behavior of the noisy coefficients $u_i^T b = u_i^T b^{exact} +$

$u_i^T e$ as well as the size of singular values. We only want to include the SVD component when its contribution to the real solution is thought to outweigh its noise. Selective SVD further extends the idea by selecting the SVD components instead of choosing a cut-off point. It keeps track of the size of $\frac{u_i^T b}{\sigma_i}$ so that small values are discarded as they are more likely to contain more noise than information [14].

2.1.2 Tikhonov Regularization

The inverse solution can also be computed by solving a least squares problem of the form

$$\min_x ||Ax - b||_2 \tag{2.1}$$

Tikhonov Regularization includes a regularization term to the original least squares problem:

$$\min_x \{||Ax - b||_2^2 + \lambda^2 ||x||_2^2\} \tag{2.2}$$

where the regularization parameter λ balances the residual term $||Ax - b||_2^2$ and the regularization term $||x||_2^2$.

We can rewrite the Tikhonov problem as a least squares problem

$$\min_x \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2 ,$$

and the solution to this least squares problem can be written as

$$x_\lambda = (A^T A + \lambda^2 I)^{-1} A^T b. \tag{2.3}$$

To see why Tikhonov regularization is effective, observe that if we substitute the singular value decomposition, $A = U\Sigma V^T$, into the expression for x_λ, we have

$$\begin{aligned} x_\lambda &= ((U\Sigma V^T)^T U\Sigma V^T + \lambda^2 I)^{-1}(U\Sigma V^T)^T b \\ &= (V\Sigma^2 V^T + V\lambda^2 V^T)^{-1} V\Sigma U^T b \\ &= V(\Sigma^2 + \lambda^2 I)^{-1} V^T V \Sigma U^T b \\ &= V(\Sigma^2 + \lambda^2 I)^{-1} \Sigma U^T b. \end{aligned}$$

Then we expand the matrix multiplications column-wise and we get:

$$x_\lambda = \sum_{i=1}^{n} \phi_i^{[\lambda]} \frac{u_i^T b}{\sigma_i} v_i$$

where the filter factors $\phi_i^{[\lambda]}$ are

$$\phi_i^{[\lambda]} = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}.$$

Notice that for moderately small values of λ (e.g. $\lambda = 10^{-3}$), the filter factor $\phi_i^{[\lambda]}$ is approximately equal to 0 for tiny singular values and approximately equal to 1 for larger singular values. It therefore acts like a filter by decreasing the effects of magnifying noise in b when divided by tiny singular values.

The advantage of Tikhonov regularization is that it can be easily implemented in large scale problems, where we only need to reformulate the matrix A into $\begin{pmatrix} A \\ \lambda I \end{pmatrix}$ and the right hand side b into $\begin{pmatrix} b \\ 0 \end{pmatrix}$. And then we can apply iterative methods on the new problem. Tikhonov regularization has similar behavior as the truncated SVD method.

2.1.3 Iterated Tikhonov Regularization

The iterated Tikhonov method applies tikhonov regularization to the residual in each iteration to further refine the obtained result. It belongs to a more general class of iterative method called iterative refinement. Iterative refinement was designed to reduce the accumulation of numerical errors when solving a liear system by iteratively correcting the solution x using its residual. Specifically, in the m^{th} iteration, three steps are performed:

1. compute the residual $r_m = b - Ax_m$,
2. solve $Ah_m = r_m$
3. add the correction $x_{m+1} = x_m + h_m$

When $b = b^{exact}$ (no noise on the right hand side) and computation has no round-off errors, the process would converge to the correct solution [20] [24].

Iterative Tikhonov adds regularization during step (2) in each iteration to reduce the effect of noise contained in b. Again we replace the original least squares problem with a penalty minimized version:

$$\min_x \{||Ax - b||_2^2 + \lambda^2 ||x - x^*||_2^2\} \tag{2.4}$$

where λ is the regularization parameter and x^* is an approximation of the real solution from prior knowledge. When such x^* is not available, it can be set to zero [3].

For Tikhonov regularization in general form, a regularization matrix is added to the minimization problem:

$$\min_x \{||Ax - b||_2^2 + \lambda^2 ||L(x - x^*)||_2^2\} \tag{2.5}$$

Here L is the regularization matrix that satisfies $\mathcal{N}(L) \cap \mathcal{N}(A) = \{0\}$ where $\mathcal{N}(L)$

and $\mathcal{N}(A)$ are null spaces of L and A. In this thesis we set L to be the identity matrix and x^* to be zero.

After obtaining a solution x_0 to the minimization problem 2.4, we then hope to get an approximation of the error for x_0 by considering another Tikhonov regularized minimization problem with the residual on the right hand side:

$$\min_h \{||Ah - r_0||_2^2 + \lambda^2 ||h||_2^2\} \tag{2.6}$$

where $r_0 = b - Ax_0$. Symbolically, we have

$$h = (A^T A + \lambda^2 I)^{-1} A^T r_0 \tag{2.7}$$

And a refined approximation of the solution is obtained by moving x_0 in the direction of the approximated error:

$$x_1 = x_0 + h$$

This repeated process of refining the solution using the residual defines the iterated Tikhonov method [9] [3]. The following algorithm describes the method:

Algorithm 1 Iterated Tikhonov

```
Initialize $x_0$ = initial guess
for $k = 0, 1, \ldots$ do
    $r_k = b - Ax_k$
    if $||r_k||_2$ <tol then
        exit
    end if
    $x_{k+1} = x_k + (A^T A + \lambda I)^{-1} A^T r_k$
end for
```

Choosing the regularization parameter λ determines how sensitive the solution is to the error e in b and how close the solution is to x^{exact}. If the regularization parameter in each iteration is the same, the method is said to be stationary, otherwise it is non-stationary. A classical choice in the non-stationary case is to use a decreasing geometric sequence of values, which has a linear convergence rate for certain adaptive choices of λ as established by Brakhage. Other choices include a nondecreasing sequence of regularization parameters proposed by Donatelli [7].

2.2 Low Precision Arithmetic

Floating point formats are engineered to store and represent numbers in computers. Most computer processors use double-precision binary floating point arithmetic, which represents floating-point numbers with 64 bits. The formats are established by IEEE, which specify the number of bits assigned to the sign, exponent, and mantissa. Some most common formats include double precision (one signed bit, 11 bit for exponent, 52 bit for mantissa), single precision (one signed bit, 8 bit for exponent, 23 bit for mantissa) and half precision (one signed bit, 5 bit for exponent, 10 bit for mantissa) as shown in Figure 2.2.

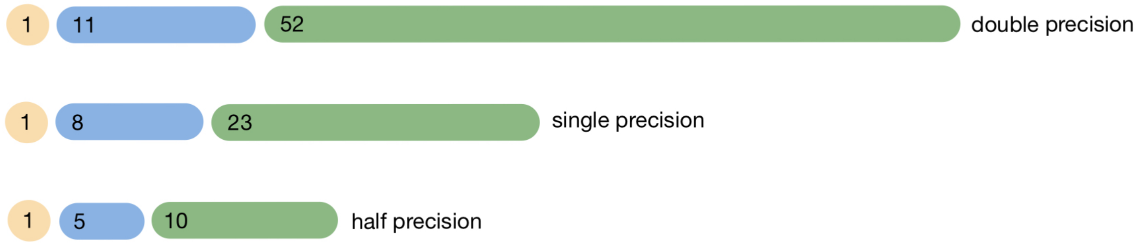

Figure 2.2: Floating point formats for double, single and half precision.

Low/mixed precision computation is becoming increasingly popular in fields such as deep learning, gaming, and other large-scale modelling. In deep learning where the number of parameters can be huge, low precision training is used to boost the performance and power efficiency of deep learning hardware [21]. Hubara et al. introduced a method to train Quantized Neural Networks using weights and activations at 1-bit precision that yields prediction accuracy comparable to their 32-bit counterparts [17]. Mixed precision versions of matrix factorization algorithms have also been developed. For example, Abdulah et al. presented a mixed-precision tile algorithm for Cholesky factorization that is 1.6X faster while maintaining necessary accuracy [1]. On the other hand, Yamazaki et al. used mixed precision to enhance the stability of CholQR by raising some crucial intermediate steps to higher precision [25].

Benefits of computing in low/mixed precision include requiring fewer resources for both processing and memory storage, as well as less power consumption and reduced computation time. For many algorithms, mixed precision has proved to achieve similar final accuracy with large speed-up and savings. In a word, mixed precision methods benefit algorithms that are limited by either computation or bandwidth [12].

Chapter 3

Modified Iterated Tikhonov

In this chapter, we go deeper into the iterated Tikhonov method to review some of its variations. At the end, we introduce our approach, which seeks to extend the research on iterative refinement in low precision to ill-posed problems using iterated Tikhonov regularization.

3.1 Replacing the Original Matrix with a Close Approximation

3.1.1 Algorithm Overview

In [8], Donatelli and Hanke introduced an iterative scheme similar to the iterated Tikhonov regularization method. In the proposed scheme, the original operator is replaced by an approximation that is close enough to the original operator but can speed up calculation due to its special properties.

Again we consider the ill-posed problem:

$$Ax = b^{exact} \tag{3.1}$$

where $A: \mathcal{X} \to \mathcal{Y}$ is a linear operator mapping from $\mathcal{X}$ to $\mathcal{Y}$.

An approximation C of A is constructed under a closeness assumption (i.e. spectral equivalence condition)

$$||(C - A)z|| \leq \rho ||Az||, \qquad z \in \mathcal{X} \tag{3.2}$$

for some $0 < \rho < \frac{1}{2}$.

The difference between this method and the Tikhonov method is that the calculation of the residual correction step is based on C instead of the original matrix A to speed up computation. Again assume that the right hand side b^{exact} is blended with noise and the noisy approximation b satisfies

$$||b - b^{exact}|| \leq \delta \tag{3.3}$$

In each step, we compute

$$h_n = \min_h \{||Ch - r_n||_2^2 + \lambda_n^2 ||h||_2^2\}, \qquad r_n = b - Ax_n \tag{3.4}$$

which is equivalent to computing $h_n = C^T(CC^T + \lambda_n^2 I)^{-1} r_n$.

Then we update the solution

$$x_{n+1} = x_n + h_n. \tag{3.5}$$

The complete algorithm is described in Algorithm 2.

Donatelli and Hanke proved that when there is no noise ($\delta = 0$), the sequence of solutions obtained by the method converges to the solution of (3.1) that is closest to the initial guess x_0.

Algorithm 2 Iterative scheme by Donatelli and Hanke

```
Initialize $x_0$, and set $r_0 = b - Ax_0$. Choose $\tau = (1+2\rho)/(1-2\rho)$ with $\rho$ from (3.2). Fix $q \in (2\rho, 1)$
while $||r_n|| > \tau\delta$, let $\tau_n = ||r_n||/\delta$ do
    $h_n = C^T(CC^T + \lambda_n^2 I)^{-1} r_n$
    where $\lambda_n$ is such that $||r_n - Ch_n|| = q_n||r_n||$, $q_n = \max\{q, 2\rho + (1+\rho)/\tau_n\}$
    $x_{n+1} = x_n + h_n$
    $r_{n+1} = b - Ax_{n+1}$
end while
```

3.1.2 Circulant Approximation and Fast Fourier Transform

For image deblurring problems, a suitable choice of C can be a circulant matrix or BCCB (block circulant with circulant blocks) matrix. Circulant matrix refers to matrices where each row is a circular shift of the previous row. It is a special type of Toeplitz matrix. One favorable property of such matrices is that they can be diagonalized as

$$C = F^H \Lambda F$$

where F is the Fourier transform matrix and is unitary (i.e. $F^H F = FF^H = I$). Both F and F^H are symmetric. Similar diagonalization exists for a BCCB matrix where the F becomes the 2D Fourier transform matrix.

Matrix vector multiplication with F is fast when using an FFT (fast fourier transform). There are many algorithms to implement FFT, one of the most commonly used is the Cooley–Tukey algorithm that gives result in less than $2n \log_2 n$ operations [6], which is a huge reduction compared with $n(2n-1)$ operations for multiplication with a generic matrix. The algorithm goes through a repeatedly divide and conquer process where the original problem is recursively broken down into sub-problems. The most well known use is the radix-2 case where the transform is divided into two transforms with size half the original transform. There are also mixed-radix cases that extends to transforms that are not a power of two.

Using a circulant matrix or BCCB matrix therefore makes the algorithm faster, especially for large scale problems where computation can be reduced significantly. In fact, the only information we need is the eigenvalues on the diagonal of Λ, which can be easily computed by applying discrete Fourier transform to the first column of the matrix and scale it by a factor of $\sqrt{n}$:

$$\mathrm{diag}(\Lambda) = \sqrt{n} F C(:,1) \tag{3.6}$$

where n is the size of A. This can be verified if we plug in $C(:,1) = F^H D_1 F(:,1)$.

Again, we can compute the eigenvalues using FFT with a time complexity of $O(n \log n)$. In contrast, a general eigenvalue algorithm, such as the QR algorithm for Hessenberg matrices, typically has a cost of $O(n^2)$ per iteration. Therefore for circulant or BCCB matrices, the computation in equation (3.4) can be performed easily and with high efficiency.

3.2 Arnoldi-based Preconditioner

Buccini et al. proposed another variant of Tikhonov regularization method using a few steps of the Arnoldi process [4]. The problem setting is a little different from the general problem 2.1 in that the matrix A is such that matrix-vector products Aw can be evaluated inexpensively while $A^T w$ cannot.

3.2.1 Arnoldi Process

Arnoldi Process is one of the Krylov subspace methods that project the problem onto the Krylov subspace $K_m(A, v) = span\{v, Av, A^2v, \ldots, A^{m-1}v\}$. It computes an orthogonal basis of K_m and can be used to find eigenvalues of the matrix A. The algorithm is described below:

Algorithm 3 Arnoldi Process

```
Initialize $||v_1||_2 = 1$
for $j = 0, 1, 2, \dots, m$ do
    $w = Av_j$
    for $i = 0, 1, 2, \dots, j$ do
        $h_{i,j} = w \cdot v_i$
        $w = w - h_{i,j}v_i$
    end for
    $h_{j+1,j} = ||w||_2$
    $v_{j+1} = w/h_{j+1,j}$
end for
```

We can see the matrices have the following relationship:

$$AV_m = V_{m+1}H_{m+1}. \tag{3.7}$$

After block operation, we can further write

$$AV_m = V_mH_m + h_{m+1,m}v_{m+1}e_m^T.$$

Multiply both sides by V_m^T on the left and we get

$$V_m^TAV_m = H_m + h_{m+1,m}V_m^Tv_{m+1}e_m^T = H_m$$

where V_m is a matrix with orthogonal columns that span the Krylov subspace formed by A and v, and H_m is an upper Hessenberg.

3.2.2 Iterated Arnoldi-Tikhonov Method

The iterated Arnoldi-Tikhonov Method (IAT) proposed by Buccini et al. follows the iterated Tikhonov scheme. At each iteration, Tikhonov regularization is added and a correction vector is calculated from the current residual as in equation (2.7). However, the original matrix A here is substituted by the p^{th} step of the decomposition obtained from the Arnoldi process as in equation (3.7). Then, the correction vector h_k becomes:

$$\begin{aligned} h_k &= A^T(AA^T + \lambda^2 I)^{-1} r_k \\ &= (V_{p+1}H_{p+1}V_p^T)^T(V_{p+1}H_{p+1}V_p^T(V_{p+1}H_{p+1}V_p^T)^T + \lambda^2 I)^{-1} r_k \\ &= (V_p H_{p+1}^T V_{p+1}^T)(V_{p+1}(H_{p+1}H_{p+1}^T + \lambda^2 I)V_{p+1}^T)^{-1} r_k \\ &= V_p H_{p+1}^T (H_{p+1}H_{p+1}^T + \lambda^2 I)^{-1} V_{p+1}^T r_k \\ &= V_p (H_{p+1}^T H_{p+1} + \lambda^2 I)^{-1} H_{p+1}^T V_{p+1}^T r_k. \end{aligned} \tag{3.8}$$

Replacing A with $V_{p+1}H_{p+1}V_p^T$ satisfies the spectral equivalence condition in equation (3.2) automatically since they are essentially the same matrix.

3.3 Replacing the Original Matrix with a Low Precision Version

In this thesis, we approximated the original matrix A with a lower precision version of itself. In other words, in the scheme by Donatelli and Hanke, the approximation C of A here is A in low precision. We used the Conjugate Gradient algorithm for Least Squares (CGLS) with Tikhonov regularization for the computation to calculate the correction vector h in each iteration, and computations are all performed in low precision. The motivations behind this are:

- Though it might require more iterations to converge, computing in low precision can potentially reduce memory costs and speed up computation.

- The major difficulty for low precision (especially half precision) to reach a good solution in such problems is that overflow can occur easily due to the limited number of bits allocated to the exponent. Roundoff errors can also accumulate due to the limited number of bits allocated to the mantissa. Iterated Tikhonov divides its iterations into multiple steps, where each step contains fewer sub-iterations. In this way the accumulation of round-off errors gets cut off when a new step starts, allowing the method to run more iterations and further refine the solution.

In both [19] and [13], the authors demonstrated with numerical experiments that the number of floating point operations per second of half and single precision is about 4× and 2× that of double precision. We use this result in the Chapter 4 when comparing the computational cost of different methods.

We now want to check the spectral equivalence condition in equation (3.2). We hope to find an upper bound for the ρ such that

$$\rho = \max_{z \in \mathcal{X}} \frac{||(C - A)z||}{||Az||} \tag{3.9}$$

where A is the original matrix stored in double precision and C is lower precision version of A. We assume A has full column rank because otherwise the null space of A would be non-empty. If we take $z \in \mathcal{N}(A)$, the denominator $||Az||$ would be 0, making ρ not bounded.

In general, when we can write $H = M^H M$ for some invertible matrix M, we have:

$$\max_x \frac{x^H G x}{x^H H x} = \max_x \frac{x^H G x}{x^H M^H M x}$$

Consider $Mx = b$, and assuming M is invertible, $x = M^{-1}b$, then we can further rewrite:

$$\begin{aligned}\max_x \frac{x^H G x}{x^H H x} &= \max_b \frac{(M^{-1}b)^H G (M^{-1}b)}{b^H b} \\ &= \max_b \frac{b^H (M^{-H} G M^{-1}) b}{b^H b} \\ &= \text{largest eigenvalue of } M^{-H} G M^{-1}\end{aligned}$$

3.3.1 General Case When A is a Real Square Matrix

With consistency of 2-norm, we have:

$$
\begin{aligned}
\rho^2 &= \max_z \frac{||(C-A)z||_2^2}{||Az||_2^2} \\
&= \max_z \frac{z^T(C-A)^T(C-A)z}{z^TA^TAz} \\
&= \text{largest eigenvalue of } A^{-T}(C-A)^T(C-A)A^{-1} \\
&= \text{largest eigenvalue of } ((C-A)A^{-1})^T(C-A)A^{-1} \\
&= (\text{largest sigular value of } (C-A)A^{-1})^2 \\
&= ||(C-A)A^{-1}||_2^2 \\
&= ||CA^{-1}-I||_2^2 \\
&= ||A^{-T}C^T-I||_2^2.
\end{aligned} \tag{3.10}
$$

The next theorem [3.3.1] is useful for our analysis.

Theorem 3.3.1. *Let $Ax = b$ and $(A+\Delta A)y = (b+\Delta b)$ where $||\Delta A|| \leq \epsilon||E||$ and $||\Delta b|| \leq \epsilon||e_b||$, and assume that $\epsilon||A^{-1}||||E|| < 1$. Then*

$$
\frac{||x-y||}{||x||} \leq \frac{\epsilon||A^{-1}||||A||}{1-\epsilon||A^{-1}||||E||}\left(\frac{||e_b||}{||b||}+\frac{||E||}{||A||}\right).
$$

We use a slight variation of the theorem above when $\Delta b = 0$ and x becomes a matrix instead of a vector. Let $AX = B$ and $(A+\Delta A)Y = B$ where $||\Delta A|| \leq \epsilon||E||$, and assume that $\epsilon||A^{-1}||||E|| < 1$. From $(A+\Delta A)(Y-X) = B-B-\Delta AX = -\Delta AX$, we get $Y - X = -(A + \Delta A)^{-1}\Delta A X$. Then we can write the relative error as

$$\begin{aligned}
\frac{||X - Y||}{||X||} &\leq \frac{||(A + \Delta A)^{-1}\Delta A||||X||}{||X||} \\
&= ||(A + \Delta A)^{-1}\Delta A|| \\
&= ||(I + A^{-1}\Delta A)^{-1}A^{-1}\Delta A|| \\
&\leq ||(I + A^{-1}\Delta A)^{-1}||||A^{-1}\Delta A|| \\
&\leq \frac{1}{1 - ||A^{-1}\Delta A||}||A^{-1}\Delta A|| \\
&\leq \frac{\epsilon||A^{-1}E||}{1 - \epsilon||A^{-1}E||}.
\end{aligned} \tag{3.11}$$

Back to the problem of finding a bound for ρ, let $Y = A^{-T}C^{T}$, we can write

$$\rho = ||Y - I||_2.$$

Here we can view this as a perturbation problem with the original system be $C^T X = C^T$ and the perturbed system be $A^T Y = C^T$. Obviously the real solution is $X = I$.

To calculate the ϵ that quantifies the perturbation amount, we start from the relationship in Frobenius norm

$$||A - C||_F \leq 2^{-p}||A||_F$$

where p represents the number of mantissa bits in the current precision. This is because for every entry in the matrix we have $|c_{ij} - a_{ij}| \leq 2^{-p}|a_{ij}|$. Therefore for the 2-norm we can derive the following relationship:

$$||A - C||_2 \leq 2^{-p}\sqrt{n}||A||_2$$

since for any matrix A of size $n \times n$, $||A||_F/\sqrt{n} \leq ||A||_2 \leq ||A||_F$.

We then rewrite it in terms of relative difference with respect to C:

$$\begin{aligned}
||A-C||_2 - 2^{-p}\sqrt{n}||C||_2 &\leq 2^{-p}\sqrt{n}(||A||_2 - ||C||_2) \\
||A-C||_2 - 2^{-p}\sqrt{n}||C||_2 &\leq 2^{-p}\sqrt{n}||A-C||_2 \\
||A-C||_2 &\leq \frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}||C||_2 \\
||A^T-C^T||_2 &\leq \frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}||C^T||_2
\end{aligned} \tag{3.12}$$

given $1-2^{-p}\sqrt{n} > 0$, which is always true for $n < 2^{20}$ for half precision (i.e. $p = 10$).

Plugging $E = C^T$ and $\epsilon = \frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}$ back into equation (3.11), we get

$$\frac{||X-Y||_2}{||X||_2} \leq \frac{\epsilon||C^{-T}C^T||_2}{1-\epsilon||C^{-T}C^T||_2} = \frac{\frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}}{1-\frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}}.$$

Therefore we finally get

$$\rho = ||Y-I||_2 = \frac{||X-Y||_2}{||X||_2} \leq \frac{\frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}}{1-\frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}}.$$

3.3.2 Case When A is Circulant/Block Circulant with Circulant Blocks (BCCB)

Circulant/Block Circulant with Circulant Blocks (BCCB) can be diagonalized by the discrete Fourier transform. Notice that a circulant matrix remains circulant in different precision levels. Additionally, when we subtract one circulant matrix from another circulant matrix, the result is still a circulant matrix. This same property holds true for BCCB matrices.

Let the matrices be diagonalized as $A = F^H D_1 F$ and $C = F^H D_2 F$. Using the special diagonolization of circulant/BCCB matricies, we can simplify the original equation as

$$\begin{aligned}\rho^2 &= \max_z \frac{||(C-A)z||_2^2}{||Az||_2^2} \\ &= \max_z \frac{||(F^H(D_2-D_1)F)z||_2^2}{||(F^H D_1 F)z||_2^2} \\ &= \max_z \frac{||(D_2-D_1)Fz||_2^2}{||D_1 F z||_2^2}\end{aligned}$$

Let $x = Fz$, then we can write

$$\begin{aligned}\rho^2 &= \max_z \frac{||(D_2-D_1)Fz||_2^2}{||D_1 F z||_2^2} \\ &= \max_x \frac{||(D_2-D_1)x||_2^2}{||D_1 x||_2^2} \\ &= \max_x \frac{x^H(D_2-D_1)^H(D_2-D_1)x}{x^H D_1^H D_1 x} \qquad (3.13) \\ &= \text{largest eigenvalue of } D_1^{-H}(D_2-D_1)^H(D_2-D_1)D_1^{-1} \\ &= \max\left\{\frac{|D_2(i,i)-D_1(i,i)|^2}{|D_1(i,i)|^2} \;:\; i = 1,2,\ldots,n\right\}\end{aligned}$$

3.3.3 Case When A is not Square but is Real

Let the singular value decomposition of A be $A = U\Sigma V^T$, we have

$$A^T A = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^T\Sigma V^T.$$

Since A has full column rank, we can write

$$\Sigma = \begin{bmatrix}\Sigma_1 \\ 0\end{bmatrix}$$

where Σ_1 is a diagonal matrix with singular values of A on the diagonal. Plug into the previous equation and we have

$$A^T A = V \begin{bmatrix} \Sigma_1^T & 0 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T = V \Sigma_1^T \Sigma_1 V^T = (\Sigma_1 V^T)^T \Sigma_1 V^T = P^T P$$

where $P = \Sigma_1 V^T$ is an invertible square matrix, and P have the same Frobenius norm and 2-norm as A. Similarly we can write $C - A = Q$ with Q being an invertible square matrix that preserves the frobenius norm and 2-norm. Now we can proceed as in the square case:

$$\begin{aligned} \rho^2 &= \max_z \frac{||(C-A)z||_2^2}{||Az||_2^2} \\ &= \max_z \frac{z^T Q^T Q z}{z^T P^T P z} \\ &= \text{largest eigenvalue of } P^{-T} Q^T Q P^{-1} \\ &= ||QP^{-1} - I||_2^2. \end{aligned} \tag{3.14}$$

Therefore similar to the square case, we have

$$\rho \leq \frac{\frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}}{1 - \frac{2^{-p}\sqrt{n}}{1-2^{-p}\sqrt{n}}}.$$

Chapter 4

Numerical Experiments

In this section, we test the modified algorithm with some numerical experiments. We then compare it with some standard method such as CGLS and Hybrid LSQR on the accuracy of the solution as well as computational cost. In each refinement step, we carefully control the number of iterations CGLS runs when computing the correction vector h because computing in low precision is more likely to overflow/underflow. Our goal is to extract as much information as possible from the current residual while avoiding occurance of overflow/underflow that would cause NaNs in the next round. The criteria for choosing when to stop the iteration are: if (1) ”Inf” occurred during the calculation of the current CGLS iteration, (2) we observe a sudden large increase in residual norm in the inner iteration (more than 50% increase), or (3) we observe a sudden large increase in residual norm in the outer iteration (more than 100% increase), we stop the iteration and return result from the previous iteration. In this way, the iterated refinement process can continue improving the solution.

4.1 Spectra Test Problem

We first tried a small spectra problem of size 64 × 64 and applied 1%, 0.1% and 10% Gaussian noise respectively to the observed right hand side b. We obtained an appropriate Tikhonov regularization parameter for the first outer iteration with the Hybrid LSQR algorithm, then we used a decreasing geometric sequence to form the regularization parameters for later iterations. We set the common ratio to be 0.8. The real solution to the problem is plotted below in Figure 4.1. And in Figure 4.2 we plot the singular values of the matrix associated with this problem. We can see the singular values are decaying very quickly and the smallest ones are very close to zero, indicating the matrix is very ill-conditioned.

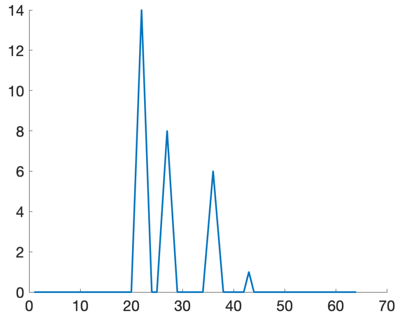

Figure 4.1: Real solution of size 64 spectra problem.

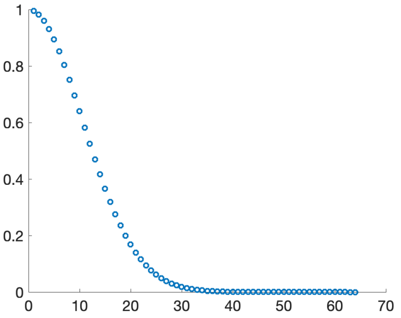

Figure 4.2: Singular values of matrix A.

4.1.1 Results

We first set the noise level to be 1% and ran the modified algorithm with the inner iteration for calculating the correction vector h in half precision. As a comparison, we show the best estimates obtained by CGLS and Hybrid LSQR in double precision. All three methods share the same Tikhonov regularization parameter.

We can see the solutions by the three methods are highly similar. Despite containing some noise as shown by the fluctuations, all three solutions accurately capture the overall pattern of the true solution by recovering the three larger "peaks." Figures 4.6 and 4.7 show the relative error norm and the relative residual norm for the three methods, with CGLS and Hybrid LSQR running in double precision, and iterated

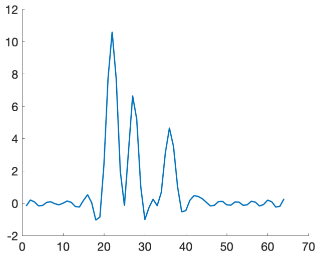

Figure 4.3: CGLS in double precision, 1% noise.

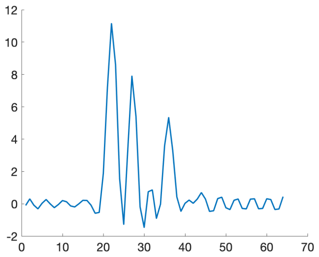

Figure 4.4: Iterated Tikhonov in half precision, 1% noise.

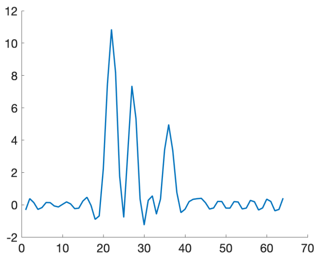

Figure 4.5: Hybrid LSQR in double precision, 1% noise.

Tikhonov running in half precision for the refinement step.

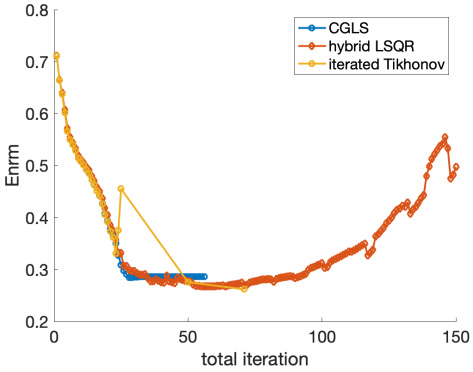

Figure 4.6: Relative error norm for spectra problem, 1% noise.

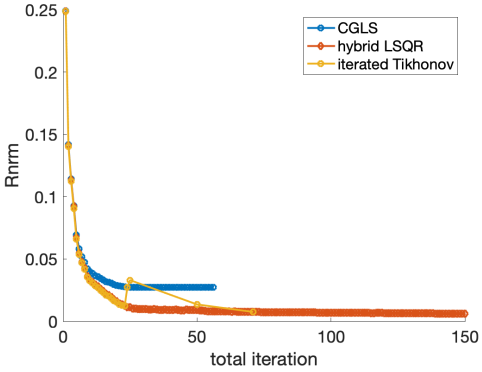

Figure 4.7: Relative residual norm for spectra problem, 1% noise.

The relative error norm of CGLS drops and then gradually converges, reaching its lowest value of 0.2841 at the 29^{th} iteration. Best solution from Hybrid LSQR has a slightly better quality with a relative error norm of 0.2675 at the 59^{th} iteration. However at the stopping iteration (30^{th} iteration), it had a much higher error norm of 0.2967. We allowed the method to continue running even after it met the stopping criteria, and no overflow occurred during the iterations. Nevertheless, the error norm increased midway through the iterations while the residual norm continued to decrease. For the iterated Tikhonov method, we are only plotting the error norm and residual norms at the end of each outer iteration. The results of the inner iterations are not displayed except for the first iteration. This is because each inner iteration is solving a different linear system to compute a correction step based on the current residual and is not directly working on the original problem. For this test problem, the method completed three outer iterations before coming to a stop, with 25, 25, 21 CGLS iterations respectively. It gave a relative error norm of 0.2615 by the time it satisfied the stopping criteria, which is the lowest of all three methods. The relative residual norms for CGLS with Tikhonov regularization and Hybrid LSQR kept decreasing smoothly as we run more iterations, with the residual norm of CGLS converging to a slightly higher value. Meanwhile, although the residual norm for iterated Tikhonov initially followed a decreasing trend and reached a value comparable to Hybrid LSQR by the end, it increased towards the end of the first outer iteration and eventually caused an overflow. We observe that the error norm exhibits a similar trend. Given the sharp increase in relative residual norm, it is natural to expect that stopping criteria (2) would halt the iteration before the error norm escalated. However, inside the inner iteration, computation is performed in low precision and the residual norm calculated with respect to the low precision version of the problem did not experience a change large enough to stop the iteration. Instead, overflow in this case helped ended the iteration, inadvertently preventing the error from further increase. And we can see later the refinement steps did make a good improvement on the result by reducing the error norms.

4.1.2 Computation Cost

We compare the computation cost of using the modified iterated Tikhonov method with correction step in half precision and directly using CGLS in double precision. In both methods, matrix-vector multiplications are the steps that require the most computations. Therefore we disregard the remaining operations such as inner product calculations and subtractions between vectors, as they are negligible in terms of computation cost. We know each CGLS iteration involves two matrix-vector multiplications. So the total number of such operations for direct CGLS is $29 \times 2 = 58$ operations in double precision, which is equivalent to $58 \times 4 = 232$ operations in half precision. While for the iterated Tikhonov method, besides the inner iteration of CGLS, in each outer iteration there is a matrix-vector multiplication in double precision when calculating the residual. So it has $(25+25+21) \times 2 = 142$ operations in half precision and two matrix-vector multiplications in double precision, making a total of 150 operations in half precision. This is less than the computation cost of direct CGLS, and achieves even better accuracy.

4.1.3 Other Noise Levels

In order to see how noise level affects the performance of the modified method, we performed similar experiments with 0.1% and 10% noise on the observed data. Figures 4.8, 4.10, and 4.12 show the solution from the three methods at noise level of 10%, and Figures 4.9, 4.11, and 4.13 show the solution from the three methods at noise level of 0.1%.

Again, the three methods yield similar results. At 10% noise level, they all managed to capture the three "peaks", though two of the peaks are not clearly distinguishable as the results at lower noise level. To take a closer look at their differences, we plotted the error norm below. The error norms at 10% noise level is plotted with number of iterations on a logarithmic scale for the sake of clarity, as Hybrid LSQR runs much more iterations compared to the other two methods.

At 10% noise level, the relative residual norm has a similar trend as the 1% noise level, with all three residual norms decreasing and CGLS being slightly larger. In terms of relative error norm, CGLS with Tikhonov regularization reached its best solution at the 22^{th} iteration with an error norm of 0.5389; Hybrid LSQR took 260 iterations to reach its best solution that has an error norm of 0.5147, the lowest of all

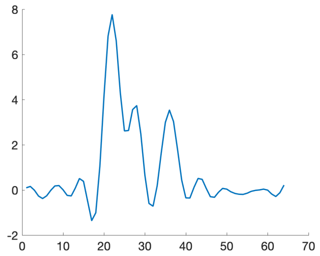

Figure 4.8: CGLS in double precision, 10% noise.

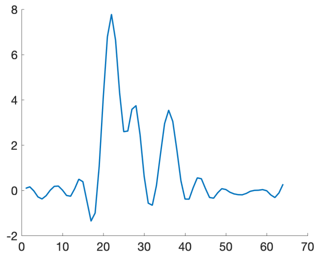

Figure 4.10: Iterated Tikhonov in half precision, 10% noise.

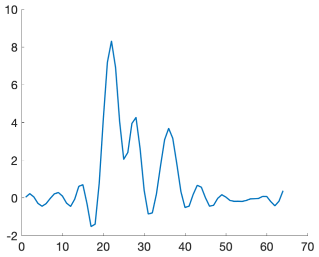

Figure 4.12: Hybrid LSQR in double precision, 10% noise.

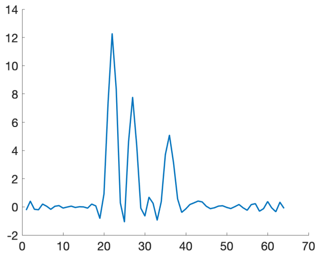

Figure 4.9: CGLS in double precision, 0.1% noise.

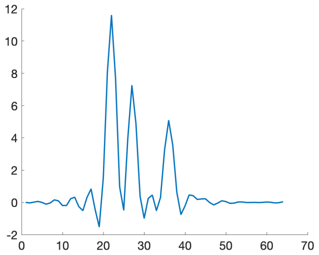

Figure 4.11: Iterated Tikhonov in half precision, 0.1% noise.

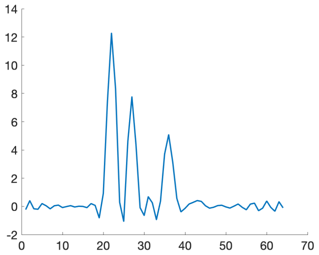

Figure 4.13: Hybrid LSQR in double precision, 0.1% noise.

three methods. However, at the stopping iteration of Hybrid LSQR (the 8^{th} iteration), the error norm is 0.5568. After 20 inner CGLS iterations, iterated Tikhonov achieved a relative error norm of 0.5373 with only one outer iteration. The method did not undergo any refinement steps due to the presence of relatively large noise in the observed right hand side, which caused the method to terminate early to avoid being influenced by noise. Interestingly, running the CGLS in half precision and double precision does not have much difference in the quality of the result. This is reasonable as the right hand side already contains a lot of noise, making the round-off errors introduced by low precision computation insignificant. Regarding computational costs, it is apparent that iterated Tikhonov takes approximately one-fourth of the computational cost required by CGLS in double precision.

At 0.1% noise level, all three methods have relative residual norm decreasing to-

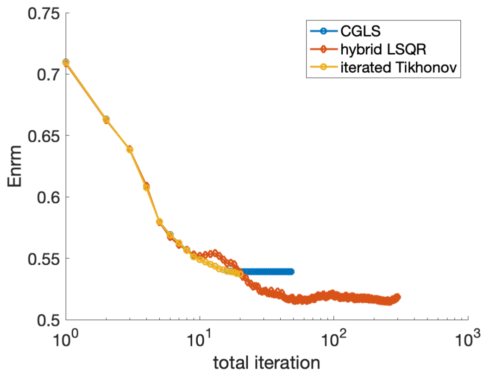

Figure 4.14: Relative error norm for spectra problem, 10% noise.

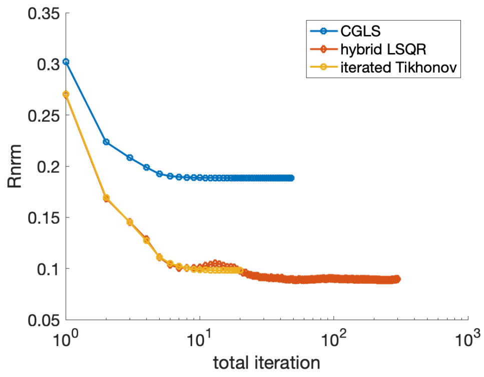

Figure 4.16: Relative residual norm for spectra problem, 10% noise.

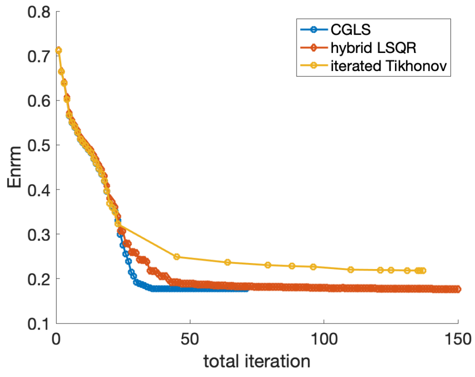

Figure 4.15: Relative error norm for spectra problem, 0.1% noise.

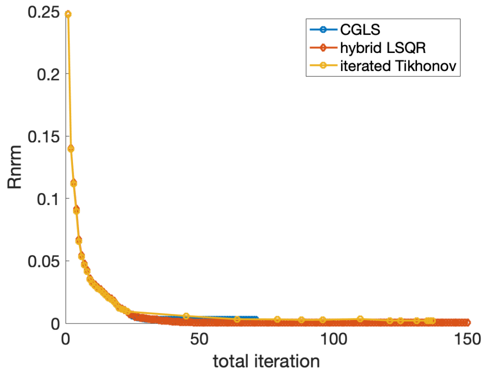

Figure 4.17: Relative residual norm for spectra problem, 0.1% noise.

wards zero. The optimal solution (achieved at the 38^{th} iteration) obtained by CGLS has a relative error norm as small as 0.1772. Meanwhile, Hybrid LSQR generated a solution of comparable quality, with an error norm of 0.1753, though at a significantly later stage (at the 181^{st} iteration). At the stopping iteration of Hybrid LSQR (44^{th} iteration), the solution had a relative error norm of 0.1920. In this test case, iterated Tikhonov is comparatively less competitive than the other two methods. This is evident from the plotted solution displayed in Figure 4.11, where the solution exhibits greater oscillations near the peaks. In terms of relative error norms, iterated Tikhonov has a higher error norm of 0.2184. We can see its error norm follows a similar decreasing trend at first and then remains above the other two methods. The iterative refinement is still effective in this case. Thirteen outer iterations are run in total, with 23, 22, 19, 15, 9, 8, 14, 11, 4, 6, 4, 1, and 1 iterations respectively. The relative error norm kept decreasing throuhgout the refinement steps. Nevertheless, iterated Tikhonov is not a good option in this case where noise is small.

4.2 Image Deblurring Test Problem

We also conducted numerical experiments on an image deblurring problem using the IRtools package [10]. The true image is a picture of the Hubble space telescope, and the observed data is corrupted by Gaussian blurring. The matrix associated with the test problem has size 4096 × 4096. We set the noise level to be 1% and common ratio for the geometric sequence of Tikhonov regularization paramter to be 0.8 as in the spectra problem. Again, we obtained the first Tikhonov parameter from the Hybrid LSQR algorithm. The true solution (the original image) and the observed right hand side (the blurred image) are plotted below in Figure 4.18 and 4.19.

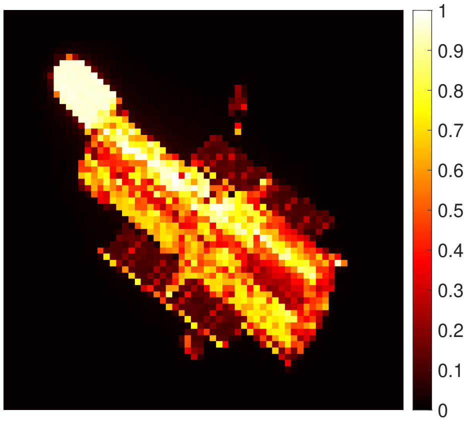

Figure 4.18: Real solution (size 64).

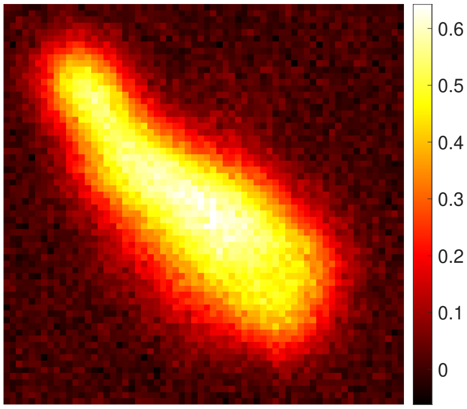

Figure 4.19: Blurred right hand side (size 64).

4.2.1 Results

We present a comparison of the optimal estimations obtained by CGLS and Hybrid LSQR in double precision with that of the modified iterated Tikhonov with the correction step in half precision. The images after deblurring using the three methods are illustrated below.

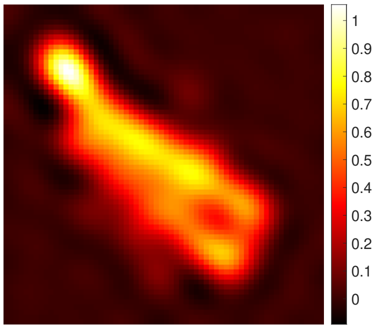

Figure 4.20: CGLS in double precision, 1% noise.

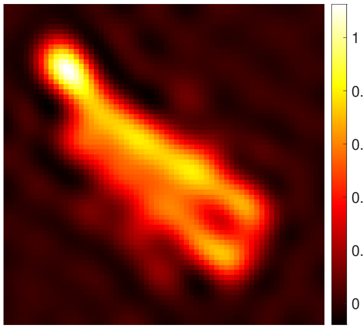

Figure 4.21: Iterated Tikhonov in half precision, 1% noise.

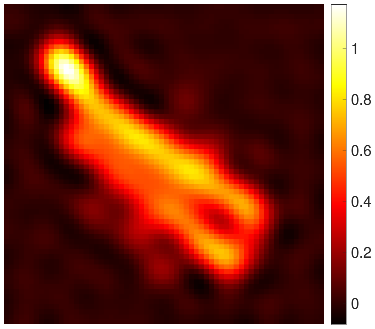

Figure 4.22: Hybrid LSQR in double precision, 1% noise.

We can see the solution given by Hybrid LSQR and iterated Tikhonov are slightly better than that of CGLS as they have sharper boundaries. However again this is because we are running the algorithms nonstop for a maximum of 300 iterations to show the best estimate. It takes Hybrid LSQR 282 iterations to reach this outcome. In fact, if we examine the solution attained when the algorithm meets its stopping criteria, its result is actually worse than that of CGLS. We plotted the relative error norm and relative residual norm below in Figures 4.23 and 4.24.

The error norms have a similar pattern as the spectra problem. The relative error norm of CGLS with Tikhonov regularization at the best estimate is 0.3685, which occurs at the 82^{nd} iteration. Hybrid LSQR attains its best solution at the 293^{rd} iteration, with a relative error norm of 0.3392. While at the stopping iteration (29^{th} iteration), the error norm is 0.3710. Again, for iterated Tikhonov, we plot the error norms and residual norms in relation to the total number of iterations, displaying

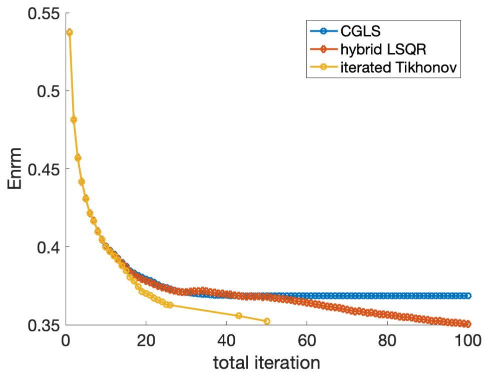

Figure 4.23: Relative error norm for deblurring problem, 1% noise.

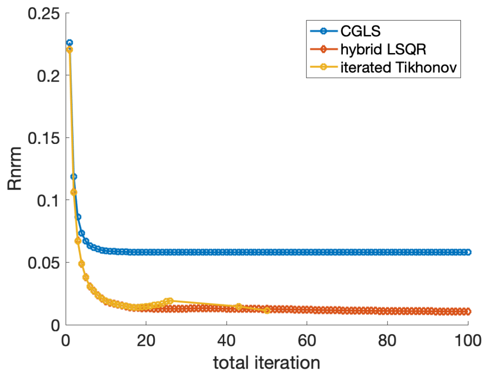

Figure 4.24: Relative residual norm for deblurring problem, 1% noise.

only the results for each outer iteration for refinement steps. The method ran 3 outer iterations before meeting the stopping criteria, and the number of iterations for the correction step for each outer iteration are 26, 17, and 7 respectively. We can see that the error norm continues to decrease, and by the 50^{th} iteration, it reaches a relative error norm of 0.3523 and comes to a stop. This gives a deblurred image of better quality compared to those obtained by CGLS and Hybrid LSQR at their stopping positions. This is evident from Figure 4.23 that the error norm for iterated Tikhonov decreases the fastest. We observed from Figure 4.24 that at the end of the first outer iteration, the relative residual norm had a tendency to increase towards the end of the first outer iteration, just before we terminated the iteration. This behavior is in line with criterion (2), which prevents sudden changes in the residual norm. In this particular example, the criterion is effective in ensuring a smooth decrease in error norm.

4.2.2 Computation Cost

Obviously the computation cost of the modified iterated Tikhonov method is significantly lower than that of CGLS as it runs fewer iterations, and a large portion of the computation is performed in half precision. Specifically, in terms of matrix-vector multiplication which consumes the most computation, CGLS requires $82 \times 2 = 164$ operations in double precision. This is equivalent to $164 \times 4 = 656$ operations in half precision. Iterated Tikhonov takes $(26+17+7) \times 2 = 100$ operations in half precision and 2 operations in double precision, which in total is 108 operations. Compared with CGLS, the iterated Tikhonov takes significantly lower cost while maintaining a better solution.

4.2.3 Other Noise Levels

As with the spectra problem, we conducted experiments and compared the performance of the three methods with noise levels of 0.1% and 10% on the right hand side. The resulting deblurred images are presented below.

We can see at 10% noise level, none of the three methods performed well in deblurring the image to a visually clear image. We can only recognize the contour of the telescope, with no additional details. Out of the three methods, Hybrid LSQR exhibits slightly better details. More details are recovered at 0.1% noise level and the difference among the three resulting images is less obvious compared to the spectra test problem. We then examine the trend of relative error of the three methods.

At the 10% noise level, the relative error norm of CGLS with Tikhonov regularization reached its optimal value of 0.4257 at the 36^{th} iteration. Again we allow Hybrid LSQR to run a maximum of 300 iterations, and after 107 iterations, it achieved the best solution with a relative error norm of 0.3951. At the stopping iteration (the 7^{th} iteration), Hybrid LSQR had a higher error norm of 0.4329, as in previous test problems. On the other hand, iterated Tikhonov achieved a relative error norm of 0.4214 after only one outer iteration that has 14 inner CGLS iterations. The method terminated early and no refinement steps were taken, and it reached even better level of deblurring quality with less than one-fourth of the computational cost required by CGLS in double precision. We observed that after reaching the stopping criteria

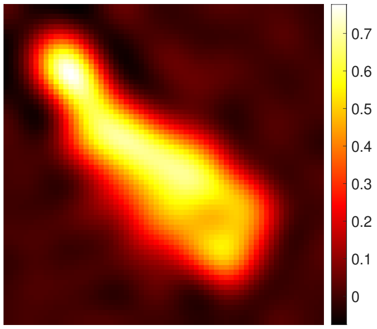

Figure 4.25: CGLS in double precision, 10% noise.

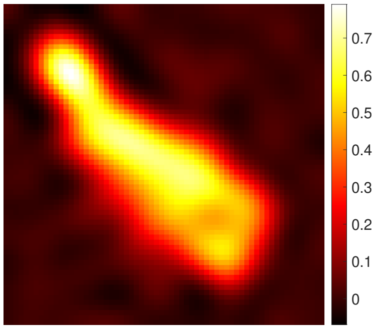

Figure 4.27: Iterated Tikhonov in half precision, 10% noise.

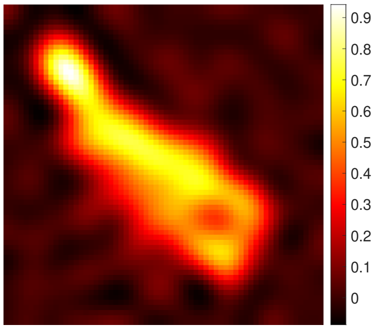

Figure 4.29: Hybrid LSQR in double precision, 10% noise.

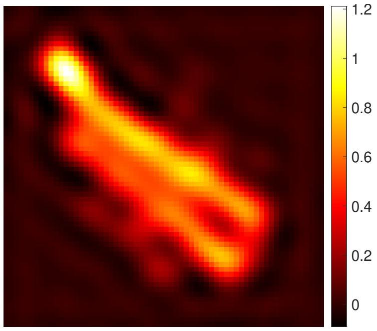

Figure 4.26: CGLS in double precision, 0.1% noise.

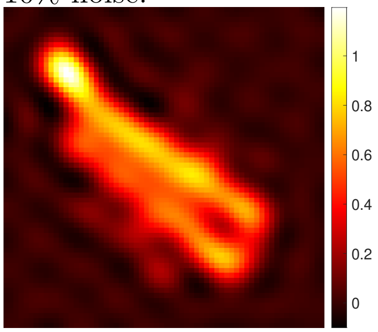

Figure 4.28: Iterated Tikhonov in half precision, 0.1% noise.

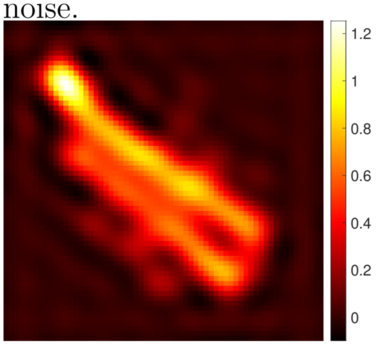

Figure 4.30: Hybrid LSQR in double precision, 0.1% noise.

for this problem, running CGLS with additional iterations leads to minimal changes in both the error norm and residual norm. In contrast, hybrid LSQR can keep on improving the results. And the stopping criteria designed for iterated Tikhonov is effective in finding a suitable time to stop the iteration before error blows up. Section 4.4 provides a more detailed discussion on this matter.

At a noise level of 0.1%, CGLS achieved the optimal solution at the 122^{nd} iteration with a relative error norm of 0.3318. Hybrid LSQR produced a solution with comparable quality, with an error norm of 0.3265 at best iteration (the 282^{nd} iteration). At the stopping iteration (the 53^{rd} iteration), Hybrid LSQR stopped with a relative error norm of 0.3400. Unlike in the spectra test problem, iterated Tikhonov

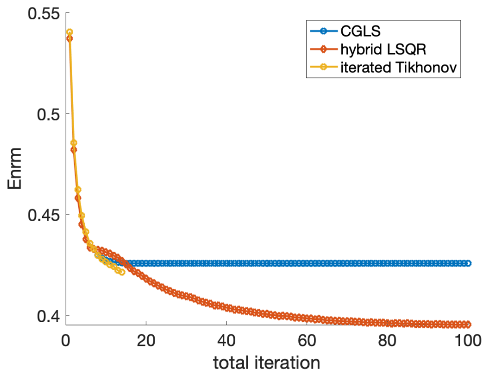

Figure 4.31: Relative error norm for deblurring problem, 10% noise.

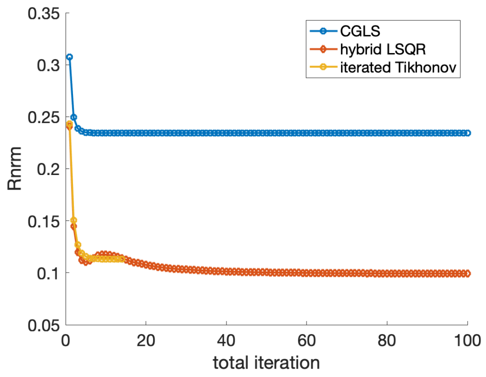

Figure 4.33: Relative residual norm for deblurring problem, 10% noise.

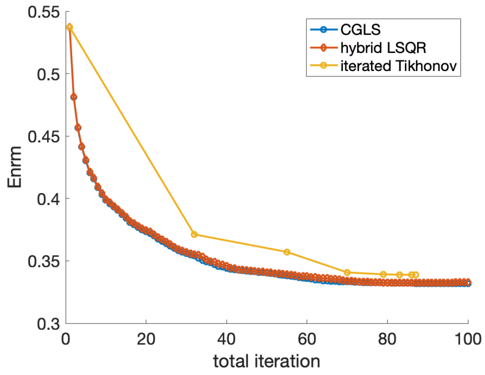

Figure 4.32: Relative error norm for deblurring problem, 0.1% noise.

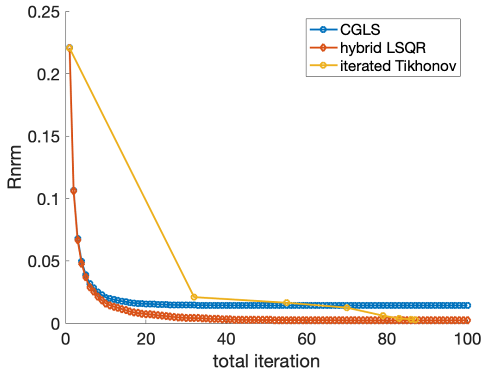

Figure 4.34: Relative residual norm for deblurring problem, 0.1% noise.

showed comparable deblurring quality to the other two methods in this particular case. From Figure 4.32 we can see the resulting error norm is very close to that of CGLS and LSQR. The relative error norm at the stopping iteration, which is also the best solution, was 0.3388. It ran eight outer iterations, with corresponding inner CGLS iterations of 1, 31, 23, 15, 9, 4, 3, and 1 respectively. Again, the computation cost of iterated Tikhonov is significantly lower than that of CGLS as it is running fewer CGLS iterations and the majority of the calculation is done in lower precision. Compared with the poor solution at low noise level for the spectra problem, one possible explanation for iterated Tikhonov being able to get to a relatively good solution in this case is the considerably larger size of the problem. Even with double precision, round-off errors accumulate more significantly in larger problems. The iterative nature of iterated Tikhonov which uses residual vectors calculated from the original matrix during refinement steps enables it to recalibrate at every iteration. This could contribute to its effectiveness.

4.3 Impact of matrix size

We hope to further investigate the relationship between the size of the matrix and the quality of the solution. As we have already observed a drop in quality in the small spectra problem and a relatively good quality in the larger image deblurring problem at low noise level.

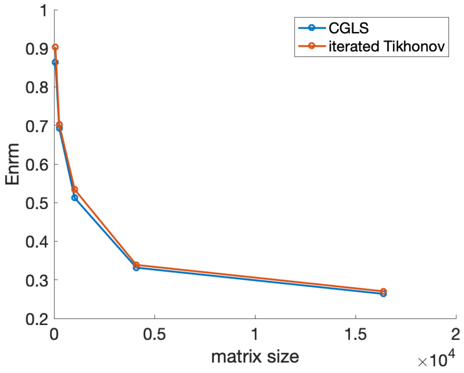

Figure 4.35: Relative error norm for deblurring problem, 0.1% noise.

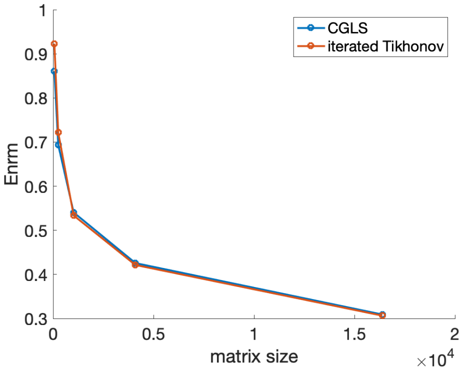

Figure 4.36: Relative residual norm for deblurring problem, 10% noise.

From the plot we can see as the matrix size gets bigger, the performance of iterated Tikhonov becomes closer or sometimes even better than that of CGLS. However at small matrix size, iterated Tikhonov can have poor performance compared with CGLS, especially when the noise level is small. This gap between CGLS and iterated Tikhonov persists until the matrix size gets sufficiently large, and the gap tends to narrow more quickly for problems with higher levels of noise.

4.4 Sensitivity to the Stopping Criteria

In this section we hope to look at the criterion (2), (3) in the stopping criterion. Most of the time the two criterion does not change the behavior of the solution too much. They are expected to help the solution improve more smoothly through the iterative refinements and force the method to come to a stop when the solution is moving in a not helpful way.

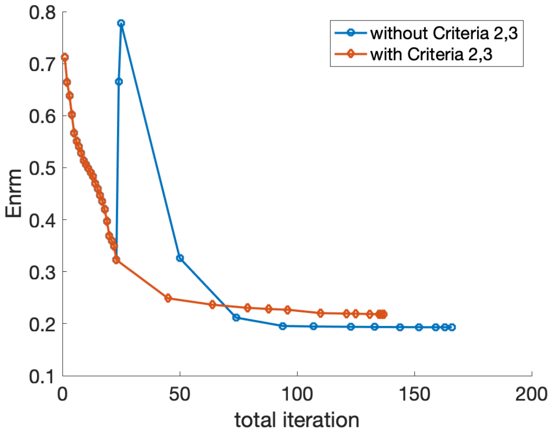

Figure 4.37: Comparing relative error norm for spectra problem, 0.1% noise, with and without criterion 2,3.

In the small spectra problem, the method is able to steadily enhance the solution with criteria (2), (3). However it is possible for the error norm to decrease further if the criteria were not in place. The convergence without criteria (2), (3) is unstable. In fact, the error norm may even surpass its value at the beginning during the iterations. Yet in the end it achieved a relative error norm that is better, though still not comparable with standard methods.

We also tested this on the larger image deblurring problem and examined the convergence trend for the problem at noise level 0.1% and 1%. We did not include the results for a noise level of 10% because the method already halted by the residual norm satisfying the stopping criteria before criteria (2) and (3) could be applied. We can see in the figures below that the criteria make the convergence more stable and could potentially lead to a better result as in Figure 4.38, or a more stable convergence as in Figure 4.39.

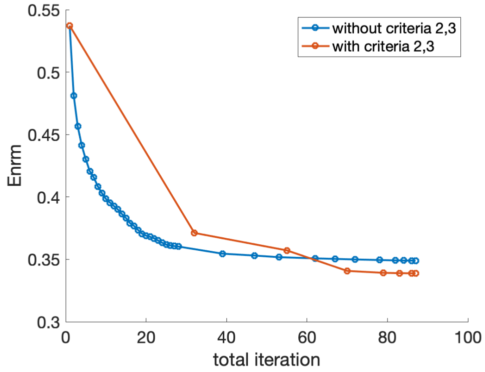

Figure 4.38: Comparing relative error norm for deblurring problem, 0.1% noise, with and without criterion 2,3.

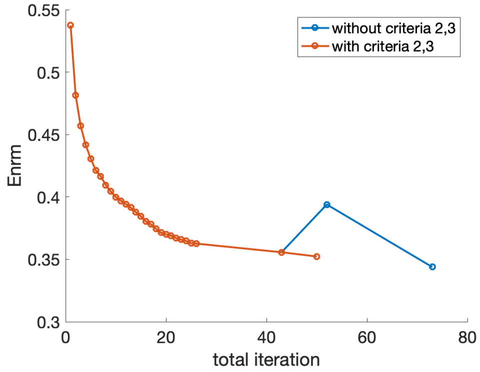

Figure 4.39: Comparing relative residual norm for deblurring problem, 1% noise, with and without criterion 2,3.

4.5 Sensitivity of the Regularization Parameter

We hope to investigate the sensitivity of the iterated Tikhonov method to the choice of regularization parameter. We tested on the image deblurring test problem used in Section 4.1.1, and chose various parameter values ranging from 0 to 1. We did not choose a parameter choice larger than 1 because both its infinity norm and 1-norm has value 1, meaning the maximum of entry's absolute value is no larger than 1. We want to avoid choosing a regularization parameter that is too large as it may overshadow the original matrix. In Figure 4.40, we plotted the resulting relative error norms corresponding to each Tikhonov parameter, with the red dot indicating the solution obtained by CGLS in double precision as a comparison.

We notice that the choice of regularization parameter does impact the quality of the solution, but the impact is not significant in a reasonable neighbourhood around the suitable parameter. Interestingly, the Tikhonov parameter determined by Hybrid

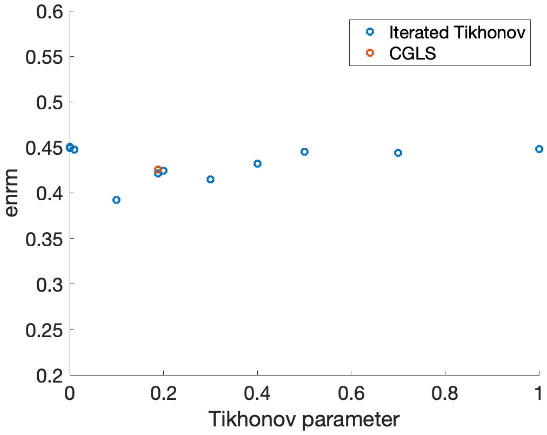

Figure 4.40: Relative error norm vs Tikhonov parameter.

LSQR is not the optimal choice. It appears that a smaller Tikhonov parameter than the one chosen by Hybrid LSQR provides a better solution. A possible explanation is that low-precision computations can easily overflow, forcing the iteration to terminate when NaNs occur. This can have a similar impact as regularization, though in a "passive" manner. As a result, the extra regularization from Tikhonov regularization could have less regularization power, making a parameter less than the regularization parameter selected by methods in double precision a better choice.

Chapter 5

Concluding Remarks

In this thesis, we explored the use of the iterated Tikhonov method for solving inverse problems with noisy right-hand sides, using low precision computation for the correction step. Our approach is inspired by the iterative scheme proposed by Donatelli and Hanke in [8], which replaces the original matrix with a computationally efficient approximation that is sufficiently close to the original one. In our case, the approximated matrix is the one in low precision. We first derived a bound for the spectral equivalence condition of matrices at different precision levels that is part of the stopping criteria in the scheme. Then, we conducted numerical experiments on a small spectra signal deconvolution test problem as well as an image deblurring problem with varying levels of noise. Our results show that most of the time, the method achieves similar results as direct CGLS with Tikhonov regularization and Hybrid LSQR, but with significantly lower computational cost.

If the noise level is sufficiently high, iterated Tikhonov may terminate after the first outer iteration and does not go through any refinement steps. This is equivalent to directly running CGLS in low precision. Despite this, the resulting solution is of similar quality to that obtained through high-precision CGLS. This is probably because compared to the noise already present in the observed data, accumulation of round-off errors from low-precision computation becomes less significant.

When the noise level is low, iterated Tikhonov is able to run more refinement steps to improve the solution. However, for small problems, the solution obtained from iterated Tikhonov can be worse than other high-precision methods. This is reasonable as using low precision computation naturally introduces more errors, which is the major source of error given the noise is small. Furthermore, we observed that the method can potentially converge to a better solution without stopping criteria (2) and (3), though in an unstable way. This is likely due to the fact that in small problems, we can allow the solution to temporarily veer in the wrong direction as we can easily steer it back at the end of the iteration using residual with respect to the original matrix as a calibration.

While for larger problems, iterated Tikhonov is capable of producing results of similar quality as other methods. This is because truncation errors are inevitable in computation, and they accumulate more rapidly with a larger matrix size, even in high-precision calculations. However, despite performing the computation of correction vectors in low precision, iterated Tikhonov adjusts the refinement direction in each step by calculating the residual in high precision. This approach allows iterated Tikhonov to achieve similar quality results by performing continuous refinement steps. Furthermore, the stopping criteria are much more effective in this case and criteria (2) and (3) do help in ensuring a stable and good convergence. They are able to stop the iteration before overfitting takes place.

The major advantage of the iterated Tikhonov method is that it requires lower computational cost while still producing solutions as good as those of standard methods like CGLS and Hybrid LSQR. This method bypasses the challenge of overflow/underflow issues that can occur when running computations in low precision by dividing the original iteration into refinement steps. The idea is to trade accuracy for computational efficiency during the refinement steps and recalibrate the direction of solution between iterations, and thereby achieving satisfactory results at lower cost.

One potential problem for the iterated Tikhonov method is that it relies on other techniques to determine a suitable Tikhonov regularization parameter to begin with. So does CGLS. But one interesting aspect about iterated Tikhonov is that it is not as sensitive to the choice of regularization parameter as CGLS computed in double precision. This is due to the fact that half precision computation is highly susceptible to overflow. Therefore, even without the regularization parameter, the iteration terminates much quicker than in double precision, which unintentionally serves as a form of early stopping regularization.

Some future work includes running more experiments on machines that support low precision to test the actual reduction in computation time. We also hope to explore more effective methods for selecting the regularization parameter, including implementing a low-precision version of hybrid LSQR.

Bibliography

[1] Sameh Abdulah, Hatem Ltaief, Ying Sun, Marc G Genton, and David E Keyes. Geostatistical modeling and prediction using mixed precision tile cholesky factorization. In *2019 IEEE 26th international conference on high performance computing, data, and analytics (HiPC)*, pages 152–162. IEEE, 2019.

[2] Giovanni S Alberti, Habib Ammari, Bangti Jin, Jin-Keun Seo, and Wenlong Zhang. The linearized inverse problem in multifrequency electrical impedance tomography. *SIAM journal on imaging sciences*, 9(4):1525–1551, 2016.

[3] Alessandro Buccini, Marco Donatelli, and Lothar Reichel. Iterated tikhonov regularization with a general penalty term. *Numerical Linear Algebra with Applications*, 24(4):e2089, 2017.

[4] Alessandro Buccini, Lucas Onisk, and Lothar Reichel. An arnoldi-based preconditioner for iterated tikhonov regularization. *Numerical Algorithms*, pages 1–23, 2022.

[5] Erin Carson and Nicholas J Higham. Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM Journal on Scientific Computing*, 40(2):A817–A847, 2018.

[6] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[7] Marco Donatelli. On nondecreasing sequences of regularization parameters for nonstationary iterated tikhonov. *Numerical Algorithms*, 60:651–668, 2012.

[8] Marco Donatelli and Martin Hanke. Fast nonstationary preconditioned iterative methods for ill-posed problems, with application to image deblurring. *Inverse Problems*, 29(9):095008, 2013.

[9] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[10] Silvia Gazzola, Per Christian Hansen, and James G Nagy. IR tools: a MATLAB package of iterative regularization methods and large-scale test problems. *Numerical Algorithms*, 81(3):773–811, 2019.

[11] Davis Gilton, Greg Ongie, and Rebecca Willett. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 6:328–343, 2019.

[12] Dominik Göddeke, Robert Strzodka, and Stefan Turek. Performance and accuracy of hardware-oriented native-, emulated-and mixed-precision solvers in fem simulations. *International Journal of Parallel, Emergent and Distributed Systems*, 22(4):221–256, 2007.

[13] Azzam Haidar, Panruo Wu, Stanimire Tomov, and Jack Dongarra. Investigating half precision arithmetic to accelerate dense linear system solvers. In *Proceedings of the 8th workshop on latest advances in scalable algorithms for large-scale systems*, pages 1–8, 2017.

[14] Per Christian Hansen. *Discrete Inverse Problems: Insight and Algorithms*. SIAM, 2010.

[15] A. Neubauer Heinz Werner Engl, Martin Hanke. *Regularization of Inverse Problems*. Springer Dordrecht, 1996.

[16] Nicholas J. Higham and Theo Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31:347–414, 2022.

[17] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1):6869–6898, 2017.

[18] Joseph B. Keller. Inverse problems. *The American Mathematical Monthly*, 83(2):107–118, 1976.

[19] Piotr Luszczek, Jakub Kurzak, Ichitaro Yamazaki, and Jack Dongarra. Towards numerical benchmark for half-precision floating point arithmetic. In *2017 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–5. IEEE, 2017.

[20] Cleve B Moler. Iterative refinement in floating point. *Journal of the ACM (JACM)*, 14(2):316–321, 1967.

[21] Xiao Sun, Naigang Wang, Chia-Yu Chen, Jiamin Ni, Ankur Agrawal, Xiaodong Cui, Swagath Venkataramani, Kaoutar El Maghraoui, Vijayalakshmi (Viji) Srinivasan, and Kailash Gopalakrishnan. Ultra-low precision 4-bit training of deep neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1796–1807. Curran Associates, Inc., 2020.

[22] Zhigang Sun and Eva M Sevick-Muraca. Inversion algorithms for particle sizing with photon migration measurement. *AIChE journal*, 47(7):1487–1498, 2001.

[23] Eduardo Tondin Ferreira Dias, Hugo Vieira Neto, and Fábio Kurt Schneider. A compressed sensing approach for multiple obstacle localisation using sonar sensors in air. *Sensors*, 20(19):5511, 2020.

[24] JH Wilkinson. Rounding errors in algebraic processes. 1965.

[25] Ichitaro Yamazaki, Stanimire Tomov, and Jack Dongarra. Mixed-precision cholesky qr factorization and its case studies on multicore cpu with multiple gpus. *SIAM Journal on Scientific Computing*, 37(3):C307–C330, 2015.

[26] Yongchao Zhang, Jiawei Luo, Jie Li, Deqing Mao, Yin Zhang, Yulin Huang, and Jianyu Yang. Fast inverse-scattering reconstruction for airborne high-squint radar imagery based on doppler centroid compensation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17, 2021.