Stabilization of GMRES and Convergence Analysis of Inner-Iteration Preconditioned GMRES for Least Squares Problems

by

Zeyu LIAO

Dissertation

submitted to the Department of Informatics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

The Graduate University for Advanced Studies, SOKENDAI
September 2022

Declaration of Authorship

I, Zeyu LIAO, declare that this thesis titled, 'Stabilization of GMRES and Convergence Analysis of Inner-Iteration Preconditioned GMRES for Least Squares Problems' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Zeyu LIAO

Date: 2022/09/12

The Graduate University for Advanced Studies, SOKENDAI

Abstract

Department of Informatics
School of Multidisciplinary Sciences

Doctor of Philosophy

by Zeyu LIAO

This thesis is devoted to the study of the generalized minimal residual methods (GMRES) for least squares problems. GMRES is a robust and efficient Krylov subspace iterative solver for nonsymmetric systems of linear equations. In this thesis, we apply GMRES to least squares problems which arise from many applications in science and engineering, etc. The application of GMRES to least squares problems has been studied, but there are still many mysteries and interesting phenomenon about it, which motivate this thesis. I would like to use three 's' to introduce this thesis, stability, speed, and size. We improved the stability of GMRES, and analyzed the convergence for the preconditioned system which could speed up iterations, and extended the techniques to problems which contain many right-hand sides.

Chapter 1 gives the background of this thesis. Chapter 2 introduced basics and notations. From Chapter 3, we propose and analyze methods to improve GMRES for least squares problems.

At first, we consider using the right-preconditioned GMRES (AB-GMRES) for obtaining the minimum-norm solution of inconsistent underdetermined systems of linear equations. Morikuni (Ph.D. thesis, 2013) showed that for some inconsistent and ill-conditioned problems, the iterates may diverge. This is mainly because the Hessenberg matrix in the GMRES method becomes very ill-conditioned so that the backward substitution of the resulting triangular system becomes numerically unstable. We propose a stabilized GMRES method based on solving the normal equations corresponding to the above triangular system using the standard Cholesky decomposition. This has the effect of shifting upwards the tiny singular values of the Hessenberg matrix which lead to an inaccurate solution. This finding seems to contradict the common sense that the normal equations are not suitable for solving ill conditioned problems, since the problem would become more ill conditioned. We presented a theorem to illustrate why the system can become better conditioned using normal equations in the presence of rounding errors and also analyzed the importance of the consistency which is ensured by the

normal equations. We analyzed the structure of the noise due to double precision arithmetic, which helps to understand how the stabilized method works. We compared our method with many existing methods, such as TSVD (Truncated Singular Value Decomposition), Tikhnov regularization and RR(Range Restricted)-GMRES, etc. Numerical experiments show that the proposed method is robust and efficient, not only for applying AB-GMRES to underdetermined systems, but also for applying GMRES to severely ill-conditioned range-symmetric systems of linear equations.

Next, we explain the super-linear convergence of the inner-iteration preconditioned GMRES method for least squares problems. Inner-iteration preconditioning is a very fascinating technique which could speed up the convergence by increasing the steps of inner-iteration. Existing error bounds are usually exponent of the spectral radius, which under logarithmic function is linear, and cannot illustrate the super-linear convergence of the method. Increasing the steps of inner-iteration will cluster the eigenvalues of the preconditioned coefficient matrix. By considering the effect of clustered eigenvalues of the preconditioned coefficient matrix, we found that eigenvalues which are close to the center help to quickly diminish the residual to a tiny level. We show that the theoretically predicted convergence behavior matches numerical experiment results. In the analysis, we assume that the preconditioned matrix is diagonalizable, but we hope extend the analysis to cases where the preconditioned coefficient matrix contains Jordan blocks in future.

Finally, we consider using the block GMRES to solve least squares problems with multiple right-hand sides. This generates the Krylov subspace and updates the QR decomposition for the Hessenberg matrix block-wise. The Block GMRES requires a larger Krylov subspace to converge than the GMRES. However, the total CPU is reduced due to efficient memory access, and the decrease of the number of iterations per right-hand side. Further, we propose combining the block GMRES method with block-wise inner-iteration preconditioning to reduce the number of iterations. Numerical experiments show that the proposed method is efficient compared to the block GMRES. We also gave some conjectures in Appendix B for the grade of block GMRES.

In conclusion, this thesis proposes a stabilized method to ensure the stability when GMRES suffers severe ill-conditioning and analyzes why the method works. Then, it illustrates the super-linear convergence of the inner-iteration preconditioned GMRES method for least squares problems, which is not only a new way to analyze the convergence but also can help to design good preconditioners. Finally, it extended the method to solve for many right-hand sides simultaneously block wise, which saves even more CPU time and leads to the research on the theory for the block case in the future.

Acknowledgements

Firstly, I would like to express my sincere gratitute to my supervisor Professor Ken HAYAMI, who guided me from the beginning to the completion of this thesis. He enhanced my basic knowledge, showed me the essence of the problems, gave me interesting problems and also powerful skills when I suffered bottlenecks. His patience and rigorous logic gave me a good model as a researcher. His trust made me more confident and could focus on research. I would like to thank him for a great deal of his time and effort to this work. He made me understand that the essence of education is love. Thus, I would also thank God for letting me meet such a good supervisor.

I would like to acknowledge my subadvisor Professor Keiichi MORIKUNI, who also spent a great deal of time on this thesis. His meaningful questions made my research broader. I learned a lot from his revisisions on my paper.

I would like to thank Professor Junfeng YIN, who is my master course supervisor and provided the chance to have internship at NII.

I would like to acknowledge my subadvisor Professor Yuji NAKATSUKASA for supervising me for nearly half a year, informing me the trend of numerical linear algebra and encouraging me to make friends at international meetings, which gave me clues to analyze the convergence in this research.

I acknowledge my second supervisor Professor Takeaki UNO for considerable support to complete my PHD career for two years.

I acknowledge my subadvisors Professor Gene CHEUNG, and my examiners Profssors Akira IMAKURA, Professor Masako KISHIDA, Professor Akihiro SUGIMOTO and Professor Ryota KOBAYASHI for valuable advice and comments.

I appreciate researchers who visited the Nation Institute of Informatics (NII) and gave presentations and had discussions, in particular, Professor Miroslav Rozložník, Professor Zhongzhi BAI, Professor Huaian DIAO, Professor Shuxin MIAO, Professor José Mas, Professor Michael Ng and Professor Per Christian Hansen.

I acknowledge former docotral student Doctor Kota SUGIHARA for long years' company. Having a friend in the same area is lucky. I would like to thank Doctor Ning ZHENG for various help both in life and research. I would also like to thank Doctor Qi XUE who took care of me for a night in Valencia when I suffered kidney stones.

I thank other visitors to NII, Doctor Yasunori AOKI, Doctor Quanyu DOU, Master Xiongfeng SONG, Master Zhixuan HUANG, Master Yishu DU, Master Püvi Verner and Master Tahitoa Arbelot and Master Dongfa MA.

I am grateful to my friends in NII, Tongji University and in life.

I would like to thank Dr. Hiroshi Murakami and Professor Lothar Reichel for valuable comments.

I acknowledge financial support by the Japan Student Service Organization.

I acknowledge the NII staff for caring for my life as a doctoral student.

I acknowledge my Japanese teachers Ms. Kazumi TAJIMA and Ms. Naoko ADACHI for teaching me patiently.

Finally, I express my gratitude to my parents and my sister, who always encourage me and support me, for their invaluable constant love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter first gives backgrounds of numerical linear algebra and linear least squares problems. Then it gives motivations of this thesis, and then ends with objectives and organizations of the thesis.

1.1 Backgrounds

Numerical linear algebra is a study which combines finite precision computers with linear algebra. It includes building models for the real world and solving the model by algorithms. Thus, it is a sub-field of numerical analysis and does not have a clear boundary with applied mathematics. The core of numerical linear algebra are algorithms. Algorithms usually have a form in linear algebra, and can be implemented on finite precision computers, which gives approximate solutions. Hence, properties of algorithms such as convergence are important subjects in numerical linear algebra.

Solving linear least squares problems is not only an important part of numerical linear algebra but there are also an underlying demand in many fields across science and engineering, such as tomography, optimization, statistics (linear regression), machine learning (support vector machine), geodetics (full wave form inversion), computational finance (option pricing), data mining, image and signal processing and curve fitting, which have requirements of solving least squares problems. Thus, designing robust and efficient methods for computing least squares solutions is very important. Traditional methods for solving least squares problems are direct methods such as the Cholesky

factorization, QR factorization, and the singular value decomposition. These methods are efficient for solving small dense problems.

However, for large-scale problems, the factorization or decomposition process will become very heavy. Unless one finishes the factorization or decomposition process until the size of the problem, one can not get the solution. Moreover, many problems arising from typical applications have a sparse structure, which means most of the entries are zero. The factorization or decomposition process usually destroys the sparsity of problems, and fails to take advantage of the sparsity, which can be used in matrix vector multiplications and save storage. Hence, it is difficult to apply direct methods to large and sparse problems because of time and storage space limitations.

Iterative methods make use of the sparsity when applying matrix vector multiplications, and give approximate solutions by an iteration process, and one can stop the iteration process when one is satisfied with the accuracy of the solution. Iterative methods are easier to implement on parallel computers than direct methods. There are many well-established iterative solution methods for the least squares problems, such as the Kaczmarz method [1], the Cimmino method [2], the Jacobi and successive overrelaxation methods (JOR and SOR) [3], and the Krylov subspace iterative methods, for example, the CGLS method [4], LSQR method [5] and LSMR method [6]. These methods are developed for solving large and sparse linear least squares problems. When the problems are well-conditioned, these methods converge fast. In the ill-conditioned case, they may converge slowly or even diverge. Then, preconditioning becomes necessary to accelerate the convergence. Appropriate preconditioners need less storage and save time. For divergent cases, we need more techniques to overcome it.

The generalized minimal residual method (GMRES) [7] is a robust Krylov subspace method for solving square systems of linear equations. It searches the solution by generating a Krylov subspace. Left preconditioning (BA-GMRES) and right preconditioning (AB-GMRES) [8] are typical ways to precondition GMRES for least squares problems. As for preconditioners, we have explicit preconditioners, and implicit preconditioners which we usually call inner-iteration preconditioning[9, 10].

Moreover, there are increasing demands for solving problems with many right-hand sides. Thus, extending algorithms to the block case becomes important, so that we also need to analyze the convergence and numerical properties for the block case. These

problems and studies existed before, but the demands for the block algorithms motivated us to investigate GMRES more deeply in the context of least squares problems in this thesis.

1.2 Motivations

Consider using the right-preconditioned generalized minimal residual (AB-GMRES) method, which is an efficient method for the underdetermined least squares problems. Morikuni (Ph.D. thesis, 2013) [9] showed that for some ill-conditioned problems which contain noise, the iterates of the AB-GMRES method may diverge. The first topic of the present thesis is to understand the reason why the AB-GMRES method diverges and find strategies to overcome the divergence.

The second topic also comes from Morikuni's research on inner-iteration preconditioning. This technique has a fascinating effect of improving convergence of the GMRES method, but classical ways of estimating the convergence fail to explain the effect. Thus, we want to know what makes the inner-iteration preconditioning so effective. Moreover, we notice that both block GMRES and the inner-preconditioning technique can reduce the iteration steps. Thus, we want to research what happens when we combine them together, since the interaction between them may lead to a fascinating result.

1.3 Objectives

In order to understand what happens in the divergence of AB-GMRES, we will truncate the iteration when the divergence happens and analyze the singular values of the problem. We also track the whole iterative process to see which part fails to give a reasonable solution. For the ill-conditioned part, we try to use a well-conditioned one to approximate, and replace it. With the help of the analysis, we develop a stabilized GMRES method which has the effect of shifting upwards the tiny singular values of the matrix in the problem. After that, we compare the proposed method with Tikhonov regularization, which is a famous technique for ill-conditioned problems. We will also give the condition when the proposed method works.

To analyze the inner-iteration preconditioning, we have to analyze the linear stationary iterative methods at first, and then compute the eigenvalue distribution of the preconditioned matrix. Then, establish convergence analysis based on the assumption of the eigenvalue distribution and other conditions. Finally, we obtain a more precise description of the convergence.

Then, we need to extend the linear stationary iterative methods to the block case. Further, we present a block GMRES method by changing the QR factorization in GMRES to the block case. We combine the block GMRES and inner-iteration together, and observe the convergence versus the number of right-hand sides and other parameters.

1.4 Organization

The rest of the thesis is organized as follows. In Chapter 2, we explain the least squares problems to be solved, and define the notations. We describe the GMRES method, which is the most basic algorithm for this thesis. In Chapter 3, we present the stabilized GMRES, including the relevant theory and previous work, the algorithms and theory of the stabilized GMRES, comparison with Tikhonov regularization and other numerical experiments. In Chapter 4, we give a convergence anlysis of inner-iteration preconditioning and corresponding numerical experiments. In Chapter 5, we review previous work on methods for solving problems with many right-hand sides, the inner-iteration preconditioned block GMRES method and also numerical experiments. In Chapter 6, we conclude the thesis and propose some future work.

All the experiments in this paper were done using MATLAB R2017b in double precision, unless specified otherwise (where we extended the arithmetic precision by using the Multiprecision Computing Toolbox for MATLAB [11]), and the computer uesd was Alienware 15 CAAAW15404JP with CPU Inter(R) Core(TM) i7-7820HK (2.90GHz).

Chapter 2

Preliminaries

This chapter gives relevant concepts which are used in later chapters. It begins with the elementary notation used throughout this thesis. Then, it introduces the problems we need to solve. Finally, it presents the GMRES method.

2.1 Basics and notations

Denote the real field by $\mathbb{R}$, and the complex field by $\mathbb{C}$. The uppercase letters denote matrices, lowercase letters denote vectors or scalars. Denote the identity matrix by I, and the zero matrix by 0. The subscript of a matrix denotes its size, such as $\mathrm{I}_n \in \mathbb{R}^{n\times n}$. e_i denotes the ith column of I. The superscript T denotes transpose, such as a^{T} is transpose of a, A^{T} is transpose of A. (a, b) denotes the inner product $a^{\mathsf{T}}b$, where a and b are real vectors. Denote the diagonal matrix

$$\begin{pmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{pmatrix}$$

by $\mathrm{diag}(d_1, d_2, \ldots, d_n)$.

Let $A \in \mathbb{R}^{m\times n}$, and $C \in \mathbb{R}^{n\times n}$. min denotes minimum and max denotes maximum. Then we define the following.

The maximum number of linearly independent columns or rows of A is the rank of A, and we denote it by $\text{rank}(A)$. We say that A is full-row rank if $\text{rank}(A) = m$, and full-column rank if $\text{rank}(A) = n$. If $\text{rank}(A) < \min(m, n)$, we say that A is rank-deficient.

Denote the Euclidean vector or matrix norms by $\|\cdot\|_2$, Frobenius norms by $\|\cdot\|_F$. The condition number $\kappa_2(A) = \|A\|_2\|A^\dagger\|_2$, where $A^\dagger$ is the pseudoinverse of A.

Let $\mathcal{R}(A) = \{y = Ax | x \in \mathbb{R}^n\}$ denote the range space of A, $\mathcal{N}(A) = \{x \in \mathbb{R}^n | Ax = 0\}$ denote the null space of A. Let $S \in \mathbb{R}^n$ be a subspace and $\forall$ mean for all. Then we denote the orthogonal complement of S by

$$S^\perp = \{u \in \mathbb{R}^n | (u, v) = 0, \forall v \in S\}. \tag{2.1}$$

The subspace spanned by vectors $v_i \in \mathbb{R}^n, i = 1, 2, \ldots, k$ is denoted by

$$\text{span}\{v_1, v_2, \ldots, v_k\} = \left\{w \in \mathbb{R}^n \middle| w = \sum_{i=1}^{k} c_i v_i, c_i \in \mathbb{R}\right\}. \tag{2.2}$$

The subspace generated by $C \in \mathbb{R}^{n\times n}$ and $v \in \mathbb{R}^n$

$$\mathcal{K}_k(C, v) = \text{span}\{v, Cv, \ldots, C^{k-1}v\}. \tag{2.3}$$

is called the Krylov subspace of order k. The Krylov subspace will become invariant with increase of k, the order d which reaches the invariant subspace is called the grade, i.e. minimum value of k such that $\mathcal{K}_k(C, v) = \mathcal{K}_{k+1}(C, v)$ is the grade of d of C with respect to v.

If $C = C^\mathsf{T}$, C is symmetric, we have the following definitions. We say that C is positive definite if $x^\mathsf{T}Cx > 0, \forall x \neq 0 \in \mathbb{R}^n$, positive semidefinite if $x^\mathsf{T}Cx \geq 0, \forall x \in \mathbb{R}^n$, negative definite if $x^\mathsf{T}Cx < 0, \forall x \neq 0 \in \mathbb{R}^n$. We say C is definite if it is positive definite or negative definite.

2.2 Problems

Consider the linear least squares problem

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_2, \qquad A \in \mathbb{R}^{m\times n}, \qquad b \in \mathbb{R}^m. \tag{2.4}$$

The least squares problem (2.4) is equivalent to the normal equations

$$A^{\mathsf{T}}Ax = A^{\mathsf{T}}b. \tag{2.5}$$

In terms of the shape of the matrix A, if $m = n$, we have a square coefficient matrix. If $m < n$, we say the problem (2.4) is underdetermined, and overdetermined if $m > n$. In terms of the right-hand side b, we say the problem (2.4) is consistent if $b \in \mathcal{R}(A)$, and inconsistent if $b \notin \mathcal{R}(A)$.

In the square case, if A is full-rank, the problem is consistent. In the underdetermined case, if A is full-row rank, the problem (2.4) is consistent. If A is rank-deficient, the problem (2.4) could be inconsistent.

2.3 GMRES

The generalized minimal residual method (GMRES) [7] developed by Yousef Saad and Martin H. Schultz in 1986, is an iterative method for the numerical solution of a nonsymmetric square system of linear equations. GMRES is a generalization of the MINRES method [12] which is developed by Chris Paige and Michael Saunders in 1975.

GMRES generates a Krylov subspace, and finds the solution in the Krylov subspace by minimizing the residual. Consider the problem with square coefficient matrix

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n}, \qquad b \in \mathbb{R}^n. \tag{2.6}$$

Let x_0 be the initial solution (in all our numerical experiments, we set $x_0 = 0$), the initial residual $r_0 = b - Ax_0$. Generate the Krylov subspace with A and r_0.

$$\mathcal{K}_k(A, r_0) = \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}. \tag{2.7}$$

At each step, we try to find $z_k \in \mathcal{K}_k(A, r_0)$ and $x = x_0 + z_k$, such that the residual $r_k = b - Ax_k = b - A(x_0 + z_k) = r_0 - Az_k$ is minimized, i.e. $\min_{z_k\in\mathcal{K}_k(A,r_0)} \|r_0 - Az_k\|_2$.

The Krylov subspace expands by introducing a new vector Ax_k. The new vector is generated by the multiplication of previous vector and A. The vectors $r_0, Ar_0, \ldots, A^{k-1}r_0$

might be close to linearly dependent. Thus, we need the Arnoldi's method [13] to build a set of orthogonal basis for $\mathcal{K}_k(A, r_0)$.

Algorithm 1 Arnoldi's method

1: Choose $v_1 \in \mathbb{R}^n, \quad \|v_1\|_2 = 1,$
2: **for** $i = 1, 2, \ldots, k$ **do**
3: **for** $j = 1, 2, \ldots, i$ **do**
4: $h_{i,j} = (Av_i, v_j), \quad w_i = w_i - h_{j,i}v_j,$
5: **end for**
6: $h_{i+1,i} = \|w_i\|_2, \quad v_{i+1} = w_i/h_{i+1,i}.$
7: **end for**

In a computer, due to rounding errors, $h_{i+1,i}$ generally will not reach 0, but a tiny value. One should stop the Arnoldi's method when $h_{i+1,i}$ is near machine epsilon ϵ.

In practice, we prefer a modified version as follows.

Algorithm 2 Arnoldi-Modified Gram-Schmitt

1: Choose $v_1 \in \mathbb{R}^n, \quad \|v_1\|_2 = 1,$
2: **for** $i = 1, 2, \ldots, k$ **do**
3: $w_i = Av_i,$
4: **for** $j = 1, 2, \ldots, i$ **do**
5: $h_{i,j} = w_i^{\mathsf{T}} v_j, \quad w_i = w_i - h_{j,i}v_j,$
6: **end for**
7: $h_{i+1,i} = \|w_i\|_2, \quad v_{i+1} = w_i/h_{i+1,i}.$
8: **end for**

The two versions of the Arnoldi's method are mathematically equivalent. The differences are caused by rounding errors. Modified version is more reliable. But even Arnoldi-Modified Gram-Schmidt method can lose orthogonality. In that case, one can redo line 4-6 in Algorithm 2 again or more times, it is know that once is enough, even j from i to 1 or a random order, which is help to improve the orthogonality of the basis.

Through the orthogonalization process of Arnoldi's method, the following relation holds

$$AV_k = V_{k+1}H_{k+1,k}, \tag{2.8}$$

where $V_k = \{v_1, v_2, \ldots, v_k\}$, $V_{k+1} = \{v_1, v_2, \ldots, v_k, v_{k+1}\}$ and $H_{k+1,k} = (h_{ij}) \in \mathbb{R}^{(i+1)\times i}$. In paticular, choose $v_1 = r_0/\|r_0\|_2$, start generating the orthogonal basis. After getting the orthogonal basis $V_k = \{v_1, v_2, \ldots, v_k\}$, we can express z by the linear combination

of $\{v_1, v_2, \ldots, v_k\}$, $z = V_k y$. We minimize the residual by y on the Krylov subspace.

$$\|r\|_2 = \|b - Ax\|_2 = \|b - A(x_0 + z)\|_2 = \|r_0 - Az\|_2 = \|r_0 - AV_k y\|_2 \tag{2.9}$$

Notice that $r_0 = V_{k+1}\|r_0\|_2 e_1$, where $e_1 = (1, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^{k+1}$. Denote $\beta = \|r_0\|_2$. Then, we have the following:

$$\|r_0 - AV_k y\|_2 = \|r_0 - V_{k+1}H_{k+1,k}y\|_2 = \|V_{k+1}\beta e_1 - V_{k+1}H_{k+1,k}y\|_2, \tag{2.10}$$

$$\begin{aligned}
\|V_{k+1}\beta e_1 - V_{k+1}H_{k+1,k}y\|_2^2 &= \|V_{k+1}(\beta e_1 - H_{k+1,k}y)\|_2^2 && (2.11)\\
&= [V_{k+1}(\beta e_1 - H_{k+1,k}y)]^\mathsf{T} V_{k+1}(\beta e_1 - H_{k+1,k}y) && (2.12)\\
&= (\beta e_1 - H_{k+1,k}y)^\mathsf{T} V_{k+1}^\mathsf{T} V_{k+1}(\beta e_1 - H_{k+1,k}y) && (2.13)\\
&= (\beta e_1 - H_{k+1,k}y)^\mathsf{T} \mathrm{I}_{k+1}(\beta e_1 - H_{k+1,k}y) && (2.14)\\
&= (\beta e_1 - H_{k+1,k}y)^\mathsf{T}(\beta e_1 - H_{k+1,k}y) && (2.15)\\
&= \|\beta e_1 - H_{k+1,k}y\|_2^2. && (2.16)
\end{aligned}$$

Thus, we have

$$\|r\|_2 = \|\beta e_1 - H_{k+1,k}y\|_2. \tag{2.17}$$

The algorithm of GMRES is as follows.

Algorithm 3 GMRES

Choose $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0$, $v_1 = r_0/\|r_0\|_2$,
for $i = 1, 2, \ldots, k$ **do**
  $w_i = Av_i$,
  for $j = 1, 2, \ldots, i$ **do**
    $h_{i,j} = w_i^\mathsf{T} v_j$, $w_i = w_i - h_{j,i}v_j$,
  end for
  $h_{i+1,i} = \|w_i\|_2$, $v_{i+1} = w_i/h_{i+1,i}$
  Compute $y_i \in \mathbb{R}^i$ which minimizes $\|r_i\|_2 = \|\|r_0\|_2 e_1 - H_{i+1,i}y_i\|_2$,
  $x_i = x_0 + [v_1, v_2, \ldots, v_i]y_i$, $r_i = b - Ax_i$.
  if $\|r_i\|_2 < \epsilon\|r_0\|_2$ **then**
    stop
  end if
end for

The details of computing y_i, preconditioning of GMRES, related theories and the convergence analysis will be shown in later chapters.

Chapter 3

Stabilized GMRES method

This chapter first introduces the solution of the inconsistent underdetermined least squares problem. Second, it briefly reviews the AB-GMRES method and a related theorem. Next, it demonstrates and analyzes the deterioration of the convergence. Then, it proposes and presents a stabilized GMRES method and explains a regularization effect of the method based on the normal equations for ill-conditioned problems. Finally, numerical results for the underdetermined case and the square case are presented [14–17].

3.1 Introduction

As a motivating instance when the generalized minimal residual (GMRES) method iterates diverge due to severe ill-conditioning, consider obtaining the minimum-norm solution of the inconsistent least squares problem:

$$\min_{x\in\mathbb{R}^n} \|x\|_2, \text{such that } x \in \{\arg\min_{\xi\in\mathbb{R}^n} \|b - A\xi\|_2\} \tag{3.1}$$

where $A \in \mathbb{R}^{m\times n}$ and $b \notin \mathcal{R}(A) \subseteq \mathbb{R}^m$. Here, $\mathcal{R}(A)$ denotes the range space of A. Such problems may occur in ill-posed problems where b is given by an observation which contains noise. The problem (3.1) is equivalent to

$$(A^\mathsf{T}A)^2 v = A^\mathsf{T}b, x = A^\mathsf{T}Av, \tag{3.2}$$

and the solution can be expressed by $x = A^{\dagger}b$, where A^{T} denotes the transpose of A and $A^{\dagger}$ is the pseudoinverse of A. (See e.g. [18].)

The standard direct method for solving the least squares problem (3.1) is to use the QR decomposition. However, when A is large and sparse, iterative methods become necessary. The CGLS [4] and LSQR [5] are mathmetically equivalent to applying the conjugate gradient (CG) method to the normal equations of the first kind

$$A^{\mathsf{T}}Ax = A^{\mathsf{T}}b, \tag{3.3}$$

which is equivalent to

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_2. \tag{3.4}$$

CGLS will converge to the minimum-norm solution $x = A^{\dagger}b$, provided $x_0 \in \mathcal{R}(A^{\mathsf{T}})$ (See, e.g. [18], p. 291). However, the convergence of these methods deteriorates for ill-conditioned problems and they require reorthogonalization [8] to improve the convergence. Here, we say (3.1) is ill-conditioned if the condition number $\kappa_2(A) = \|A\|_2\|A^{\dagger}\|_2 \gg 1$. Alternatively, the LSMR [6] is mathematically equivalent to applying MINRES [12] to (3.3).

Hayami et al. [8] proposed preconditioning the $m \times n$ rectangular matrix A of the least squares problem by an $n \times m$ rectangular matrix B from the right and the left, and using the generalized minimal residual (GMRES) method [7] for solving the preconditioned least squares problems (AB-GMRES and BA-GMRES methods, respectively). For ill-conditioned problems, AB-GMRES and BA-GMRES were shown to be more robust compared to the preconditioned CGNE and CGLS, respectively. Note here that BA-GMRES works with Krylov subspaces in n-dimensional space, whereas AB-GMRES works with Krylov subspaces in m-dimensional space. Since $m < n$ in the underdetermined case, AB-GMRES works in a smaller dimensional space than BA-GMRES and should be more computationally efficient compared to BA-GMRES for each iteration. Moreover, AB-GMRES has the advantage that the weight of the norm in (3.1) does not change for arbitrary B. Thus, we mainly focus on using AB-GMRES to solve the underdetermined least squares problem (3.1). Morikuni [9] showed that AB-GMRES may fail to converge to a least squares solution in finite-precision arithmetic for inconsistent problems. We will review this phenomenon. The GMRES applied to inconsistent

problems was also studied in other papers[10, 19–22]. See e.g., [19, 22, 23] for methods for solving nearly singular systems.

In this paper, we first analyze the deterioration of convergence of AB-GMRES. To overcome the deterioration, we use the normal equations of the upper triangular matrix arising in AB-GMRES to change the inconsistent subproblem to a consistent one. In finite precision arithmetic, forming the normal equations for the subproblem will not square its condition number as would be predicted by theory. In the ill-conditioned case, the tiny singular values are shifted upwards due to rounding errors. Then, applying the standard Cholesky decomposition to the normal equations will result in a well-conditioned lower triangular matrix, which will ensure that the forward and backward substitutions work stably, and overcome the problem. Our approach using the normal equations can be considered as a case where rounding errors are beneficial [24]. We analyze why the proposed method works. Numerical experiments on least squares problems with ill-conditioned rectangular coefficient matrices (Maragal_3T to 7T [25]) show that the proposed method converges to a more accurate numerical solution than the original AB-GMRES. We also show by numerical experiments that the method is effective for applying GMRES to inconsistent range-symmetric systems with singular or severely ill-conditioned square coefficient matrices.

The rest of the paper is organized as follows. In Section 2, we briefly review AB-GMRES. In Section 3, we demonstrate the deterioration of convergence of AB-GMRES applied to underdetermined inconsistent least squares problems. In Section 4, we propose and present the stabilized GMRES method which is based on normal equations and has a regularization effect for ill-conditioned problems. We also explain why the method works by performing a rounding error analysis of the method. In Section 5, numerical experiment results for applying AB-GMRES to inconsistent underdetermined systems, and for applying GMRES to inconsistent systems with severely ill-conditioned and singular range-symmetric square coefficient matrices are presented. In Section 6, we conclude the paper.

All the experiments in this paper were done using MATLAB R2017b in double precision, unless specified otherwise (where we extended the arithmetic precision using the Multiprecision Computing Toolbox for MATLAB [11]), and the computer uesd was Alienware 15 CAAAW15404JP with CPU Inter(R) Core(TM) i7-7820HK (2.90GHz).

3.2 Deterioration of convergence of AB-GMRES for inconsistent problems

In this section, we review previous work. First, we introduce the right-preconditioned GMRES (AB-GMRES). Then, we demonstrate the deterioration of convergence of AB-GMRES for inconsistent problems. Finally, we cite a related theorem to analyze the deterioration.

3.2.1 AB-GMRES

The AB-GMRES method of Hayami et al. [8] applies the GMRES method [7] to

$$\min_{u\in\mathbb{R}_m} \|b - ABu\|_2, \quad x = Bu, \tag{3.5}$$

where $B \in \mathbb{R}^{n\times m}$ is a preconditioning matrix.

Note the equivalence between the least squares problem (3.4) and the preconditioned least squares problem (3.5).

Theorem 3.1. (Theorem 3.1 of [8])

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_2 = \min_{u\in\mathbb{R}^m} \|b - ABu\|_2$$

holds for all $b \in \mathbb{R}^m$ if and only if $\mathcal{R}(A) = R(AB)$.

Lemma 3.2. (Lemma 3.3 of [8])

$\mathcal{R}(A^\mathsf{T}) = \mathcal{R}(B) \Longrightarrow \mathcal{R}(A) = \mathcal{R}(AB)$.

Theorem 3.3. (Theorem 3.6 of [8])

If $\mathcal{R}(A^\mathsf{T}) = \mathcal{R}(B)$, *then* $\mathcal{R}(AB) = \mathcal{R}(B^\mathsf{T}A^\mathsf{T}) \Longleftrightarrow \mathcal{R}(A) = \mathcal{R}(B^\mathsf{T})$.

The convergence conditions of AB-GMRES are given as follows.

Theorem 3.4. (Theorem 3.7 of [8])

If $\mathcal{R}(A^\mathsf{T}) = \mathcal{R}(B)$, *then AB-GMRES determines a least squares solution of* $\min_{x\in\mathbb{R}^n} \|b - Ax\|_2$ *for all* $b \in \mathbb{R}^m$ *and for all* $x_0 \in \mathbb{R}^n$ *if and only if* $\mathcal{R}(A) = \mathcal{R}(B^\mathsf{T})$. *Here,* $x_0 = Bu_0$ *is the initial approximate solution of (3.5) when applying AB-GMRES.*

Let $r = b - Ax = b - ABu$. Note

$$\|r\|_2^2 = \|r|_{\mathcal{R}(A)}\|_2^2 + \|r|_{\mathcal{R}(A)^\perp}\|_2^2 = \|r|_{\mathcal{R}(A)}\|_2^2 + \|b|_{\mathcal{R}(A)^\perp}\|_2^2. \tag{3.6}$$

Here, $S^\perp$ denotes the orthogonal complement of a subspace S, and $r|_{\mathcal{R}(A)}$ is the $\mathcal{R}(A)$ component of a vector r. $r|_{\mathcal{R}(A)^\perp}$ is the $\mathcal{R}(A)^\perp$ (inconsistent) componet of the residual vector r. Thus, AB-GMRES minimizes $\|r\|_2^2$, and hence $\|r|_{\mathcal{R}(A)}\|_2^2$.

The kth iterate x_k of AB-GMRES is given by

$$x_k = x_0 + Bu_k, \tag{3.7}$$

where $u_k \in \mathcal{K}_k(AB, r_0) = \mathrm{span}\{r_0, ABr_0, \ldots, (AB)^{k-1}r_0\}$, so that $x_k = x_0 + z_k$, where $z_k \in \mathcal{K}_k(BA, Br_0) = \mathrm{span}\{Br_0, (BA)Br_0, \ldots, (BA)^{k-1}Br_0\}$. Hence, if $x_0 \in \mathcal{R}(B)$, $x_k \in \mathcal{R}(B)$.

If $\mathcal{R}(B) = \mathcal{R}(A^\mathsf{T})$, then $x_k \in \mathcal{R}(A^\mathsf{T}) = \mathcal{N}(A)^\perp$. Further, if $\mathcal{R}(B^\mathsf{T}) = \mathcal{R}(A)$, then AB-GMRES determines a least squares solution x_k, i.e., $r_k|_{\mathcal{R}(A)} = 0$, where $r_k = b - Ax_k$, and that solution x_k is the minimum Euclidean norm solution.

The algorithm is given in Algorithm 4 [8]. Here, $H_{k+1,k} = (h_{ij}) \in \mathbb{R}^{(k+1)\times k}$ and $e_1 = (1, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^{k+1}$. Algorithm 4 is said to break down when $h_{k+1,k} = 0$. See Appendix B of [10].

Algorithm 4 AB-GMRES

Choose $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0$, $v_1 = r_0/\|r_0\|_2$
for $k = 1, 2, \ldots$ **do**
  $w_k = ABv_k$
  for $j = 1, 2, \ldots, k$ **do**
    $h_{j,k} = w_k^\mathsf{T} v_j, \quad w_k = w_k - h_{j,k}v_j$
  end for
  $h_{k+1,k} = \|w_k\|_2, \quad v_{k+1} = w_k/h_{k+1,k}$
  Compute $y_k \in \mathbb{R}^k$ which minimizes $\|r_k\|_2 = \|\|r_0\|_2\, e_1 - H_{k+1,k}\, y\|_2$
  $x_k = x_0 + B[v_1, v_2, \ldots, v_k]y_k, \qquad r_k = b - Ax_k$
  if $\|A^\mathsf{T} r_k\|_2 < \epsilon\|A^\mathsf{T} r_0\|_2$ **then**
    stop
  end if
end for

To find $y_k \in \mathbb{R}^k$ that minimizes the kth residual norm $\|r_k\|_2 = \|\|r_0\|_2\, e_1 - H_{k+1,k}\, y_k\|_2$ in Algorithm 4, the standard approach computes the QR decomposition of $H_{k+1,k}$

$$H_{k+1,k} = Q_{k+1} R_{k+1,k}, \qquad R_{k+1,k} = \begin{pmatrix} R_k \\ 0^{\mathsf{T}} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}, \tag{3.8}$$

where $Q_{k+1} \in \mathbb{R}^{(k+1)\times(k+1)}$ is an orthogonal matrix and $R_k \in \mathbb{R}^{k\times k}$ is an upper triangular matrix. Then, backward substitution is used to solve a system with the coefficient matrix R_k as follows

$$\|r_k\|_2 = \min_{y\in\mathbb{R}^k} \|Q_{k+1}^{\mathsf{T}}(\beta e_1) - R_{k+1,k} y\|_2, \tag{3.9}$$

where

$$\beta = \|r_0\|_2, \quad Q_{k+1}^{\mathsf{T}} \beta e_1 = \begin{pmatrix} t_k \\ \rho_{k+1} \end{pmatrix}, \quad t_k \in \mathbb{R}^k, \quad \rho_{k+1} \in \mathbb{R}. \tag{3.10}$$

Therefore,

$$y_k = \arg_{y\in\mathbb{R}^k} \|Q_{k+1}^{\mathsf{T}}(\beta e_1) - R_{k+1,k} y\|_2 = R_k^{-1} t_k, \tag{3.11}$$

and the kth iterate is given by

$$x_k = x_0 + V_k y_k, \qquad V_k = [v_1, v_2, \ldots, v_k] \in \mathbb{R}^{n\times k}, \qquad V_k^{\mathsf{T}} V_k = I, \tag{3.12}$$

where I is the identity matrix, and $v_1, v_2, \ldots, v_k$ are the basis vectors of $\mathcal{K}_k(AB, r_0)$ defined in Algorithm 4.

From now on, we use AB-GMRES to solve (3.1) with $B = A^{\mathsf{T}}$ and $x_0 \in \mathcal{R}(A^{\mathsf{T}})$, e.g. $x_0 = 0$, which means $x_k = x_0 + z_k$, where $z_k \in \mathcal{K}_k(A^{\mathsf{T}}A, A^{\mathsf{T}} r_0)$. Hence, Theorem 3.4 guarantees the convergence in exact arithmetic even in the inconsistent case. However, in finite precision arithmetic, AB-GMRES may fail to converge to a least squares solution for inconsistent problems, as shown later.

3.2.2 AB-GMRES for inconsistent problems

In this section, we perform experiments to show that the convergence of AB-GMRES deteriorates for inconsistent problems. Experiments were done on the transpose of the matrix Maragal_3 [25], denoted by Maragal_3T etc. Table 3.1 gives the information

TABLE 3.1: Information on the Maragal matrices.

matrix	m	n	density[%]	rank	$\kappa_2(A)$
Maragal_3T	858	1682	1.27	613	1.10×10^3
Maragal_4T	1027	1964	1.32	801	9.33×10^6
Maragal_5T	3296	4654	0.61	2147	1.19×10^5
Maragal_6T	10144	21251	0.25	8331	2.91×10^6
Maragal_7T	26525	46845	0.10	20843	8.91×10^6

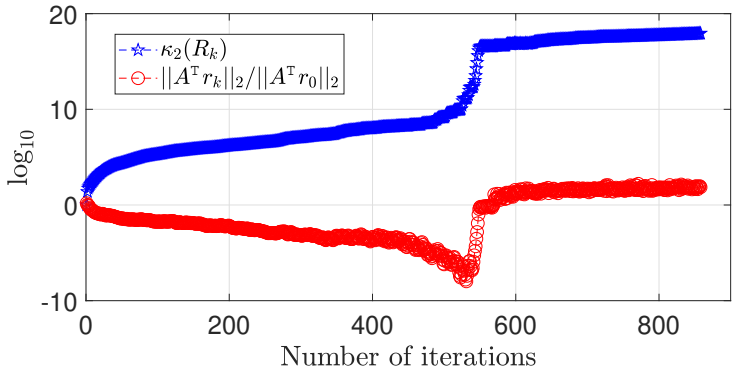


FIGURE 3.1: $\kappa_2(R_k)$ and relative residual norm versus the number of iterations for Maragal_3T.

on the Maragal matrices, including the density of nonzero entries, rank and condition number. Here, the rank and condition number were determined by using the MATLAB functions `spnrank` [26] and `svd`, respectively.

Figure 3.1 shows the relative residual norm $\|A^\mathsf{T} r_k\|_2/\|A^\mathsf{T} b\|_2$ and $\kappa_2(R_k)$ versus the number of iterations for AB-GMRES with $B = A^\mathsf{T}$ for Maragal_3T, where $r_k = b - Ax_k$, and the vector b was generated by the MATLAB function `rand` which returns a vector whose entries are uniformly distributed in the interval $(0, 1)$. Therefore, generically $b \notin \mathcal{R}(A)$ and the problem is inconsistent. Here, $\kappa_2(R_k)$=$\kappa_2(H_{k+1,k})$ holds from (3.8). The value of $\kappa_2(R_k)$ was computed by the MATLAB function `cond`. The relative residual norm $\|A^\mathsf{T} r_k\|_2/\|A^\mathsf{T} b\|_2$ decreased to 10^{-8} until the 525th iteration, and then increased sharply. The value of `cond(`R_k`)` started to increase rapidly around iterations 450–550. This observation shows that R_k becomes ill-conditioned before convergence. Thus, AB-GMRES failed to converge to a least squares solution. This phenomnenon was observed by Morikuni[9].

The reason why R_k becomes ill-conditioned before convergence in the inconsistent case will be explained by a theorem in the next subsection.

3.2.3 GMRES for inconsistent problems

Brown and Walker [19] analyzed the break-down of GMRES.

Let $b|_{\mathcal{R}(\hat{A})}$ denote the orthogonal projection of b onto $\mathcal{R}(\hat{A})$. Assume $\mathcal{N}(\hat{A}) = \mathcal{N}(\hat{A}^{\mathsf{T}})$ and $\mathrm{grade}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = k$. Here, $\mathrm{grade}(\hat{A}, \hat{b})$ for $\hat{A} \in \mathbb{R}^{m\times m}$, $\hat{b} \in \mathbb{R}^m$ is defined as the minimum k such that $\mathcal{K}_{k+1}(\hat{A}, \hat{b}) = \mathcal{K}_k(\hat{A}, \hat{b})$. Then, we have the following lemmas.

Lemma 3.5. *Assume* $\mathcal{N}(\hat{A}) \cap \mathcal{R}(\hat{A}) = \{0\}$, *and* $\mathrm{grade}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = k$. *Then,* $\mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = \hat{A}\mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})})$ *holds.*

Proof. Note that

$$\begin{aligned}\hat{A}\mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})}) &= \mathrm{span}\{\hat{A}b|_{\mathcal{R}(\hat{A})}, \hat{A}^2 b|_{\mathcal{R}(\hat{A})}, \cdots, \hat{A}^k b|_{\mathcal{R}(\hat{A})}\} \\ &\subseteqq \mathrm{span}\{b|_{\mathcal{R}(\hat{A})}, \hat{A}b|_{\mathcal{R}(\hat{A})}, \cdots, \hat{A}^k b|_{\mathcal{R}(\hat{A})}\} = \mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})}).\end{aligned}$$

$\mathrm{grade}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = k$ implies that

$$\mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = \mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = \mathrm{span}\{b|_{\mathcal{R}(\hat{A})}, \hat{A}b|_{\mathcal{R}(\hat{A})}, \cdots, \hat{A}^{k-1} b|_{\mathcal{R}(\hat{A})}\}.$$

Hence,

$$\hat{A}^k b|_{\mathcal{R}(\hat{A})} = c_0 b|_{\mathcal{R}(\hat{A})} + c_1 \hat{A}b|_{\mathcal{R}(\hat{A})} + \cdots + c_{k-1}\hat{A}^{k-1}b|_{\mathcal{R}(\hat{A})}, \quad c_i \in \mathbb{R}, i = 0, 1, 2, \cdots, k-1.$$

If $c_0 = 0$,

$$\hat{A}^k b|_{\mathcal{R}(\hat{A})} = c_1 \hat{A}b|_{\mathcal{R}(\hat{A})} + c_2\hat{A}^2 b|_{\mathcal{R}(\hat{A})} + \cdots + c_{k-1}\hat{A}^{k-1}b|_{\mathcal{R}(\hat{A})}.$$

Hence,

$$\begin{aligned}&c_1 \hat{A}b|_{\mathcal{R}(\hat{A})} + c_2\hat{A}^2 b|_{\mathcal{R}(\hat{A})} + \cdots + c_{k-1}\hat{A}^{k-1}b|_{\mathcal{R}(\hat{A})} - \hat{A}^k b|_{\mathcal{R}(\hat{A})} \\ &\qquad = \hat{A}(c_1 b|_{\mathcal{R}(\hat{A})} + \cdots + c_{k-1}\hat{A}^{k-2}b|_{\mathcal{R}(\hat{A})} - \hat{A}^{k-1}b|_{\mathcal{R}(\hat{A})}) = 0.\end{aligned}$$

Hence,

$$c_1 b|_{\mathcal{R}(\hat{A})} + c_2 \hat{A}^2 b|_{\mathcal{R}(\hat{A})} + \cdots + c_{k-1} \hat{A}^{k-2} b|_{\mathcal{R}(\hat{A})} - \hat{A}^{k-1} b|_{\mathcal{R}(\hat{A})} \in \mathcal{N}(\hat{A}) \cap \mathcal{R}(\hat{A}) = \{0\}.$$

which implies

$$\hat{A}^{k-1} b|_{\mathcal{R}(\hat{A})} = c_1 b|_{\mathcal{R}(\hat{A})} + c_2 \hat{A} b|_{\mathcal{R}(\hat{A})} + \cdots + c_{k-1} \hat{A}^{k-2} b|_{\mathcal{R}(\hat{A})}.$$

Thus,

$$\mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = \mathcal{K}_{k-1}(\hat{A}, b|_{\mathcal{R}(\hat{A})}),$$

which contradicts with $\mathrm{grade}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = k$. Hence, $c_0 \neq 0$, and

$$b|_{\mathcal{R}(\hat{A})} = d_1 \hat{A} b|_{\mathcal{R}(\hat{A})} + d_2 \hat{A}^2 b|_{\mathcal{R}(\hat{A})} + \cdots + d_{k-1} \hat{A}^{k-1} b|_{\mathcal{R}(\hat{A})} + d_k \hat{A}^k b|_{\mathcal{R}(\hat{A})}.$$

Hence,

$$\begin{aligned} \mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) &= \mathrm{span}\{b|_{\mathcal{R}(\hat{A})}, \hat{A} b|_{\mathcal{R}(\hat{A})}, \cdots, \hat{A}^k b|_{\mathcal{R}(\hat{A})}\} \\ &\subseteqq \mathrm{span}\{\hat{A} b|_{\mathcal{R}(\hat{A})}, \hat{A}^2 b|_{\mathcal{R}(\hat{A})}, \cdots, \hat{A}^k b|_{\mathcal{R}(\hat{A})}\} = \hat{A} \mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})}). \end{aligned}$$

Thus,

$$\mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = \hat{A} \mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})}).$$

□

Corollary 3.6. *Assume* $\mathcal{N}(\hat{A}) = \mathcal{N}(\hat{A}^\mathsf{T})$, *and* $\mathrm{grade}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = k$. *Then,* $\mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = \hat{A} \mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})})$ *holds.*

Proof. $\mathcal{N}(\hat{A}) = \mathcal{N}(\hat{A}^\mathsf{T})$ implies that

$$\mathcal{N}(\hat{A}) \cap \mathcal{R}(\hat{A}) = \mathcal{N}(\hat{A}^\mathsf{T}) \cap \mathcal{R}(\hat{A}) = \mathcal{R}(\hat{A})^\perp \cap \mathcal{R}(\hat{A}) = \{0\}.$$

Hence, from Lemma 3.5, Corollary 3.6 holds. □

Lemma 3.7. *Assume* $\mathcal{N}(\hat{A}) \cap \mathcal{R}(\hat{A}) = \{0\}$, $\mathrm{grade}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = k$, *and* $b \notin \mathcal{R}(\hat{A})$. *Then,* $\dim(\mathcal{K}_{k+1}(\hat{A}, b)) = k + 1$ *holds.*

Proof. Let $c_0, c_1, \ldots, c_k \in \mathbb{R}$ satisfy

$$c_0 b + c_1 \hat{A} b + \cdots + c_k \hat{A}^k b = 0.$$

Since $\mathcal{N}(\hat{A}) \cap \mathcal{R}(\hat{A}) = \{0\}$,

$$b = b|_{\mathcal{R}(\hat{A})} \oplus b|_{\mathcal{N}(\hat{A})},$$

where $b|_{\mathcal{N}(\hat{A})}$ denotes the orthogonal projection of b onto $\mathcal{N}(\hat{A})$. Hence,

$$c_0 b|_{\mathcal{N}(\hat{A})} + c_0 b|_{\mathcal{R}(\hat{A})} + c_1 \hat{A} b|_{\mathcal{R}(\hat{A})} + \cdots + c_k \hat{A}^k b|_{\mathcal{R}(\hat{A})} = 0.$$

If $c_0 \neq 0$

$$b|_{\mathcal{N}(\hat{A})} = -b|_{\mathcal{R}(\hat{A})} - \frac{c_1}{c_0} \hat{A} b|_{\mathcal{R}(\hat{A})} - \cdots - \frac{c_k}{c_0} \hat{A}^k b|_{\mathcal{R}(\hat{A})} \in \mathcal{R}(\hat{A}).$$

Hence,

$$b|_{\mathcal{N}(\hat{A})} \in \mathcal{N}(\hat{A}) \cap \mathcal{R}(\hat{A}) = \{0\}.$$

Thus, $b|_{\mathcal{N}(\hat{A})} = 0$, which contradicts $b \notin \mathcal{R}(\hat{A})$. Hence, we have $c_0 = 0$, and

$$c_1 \hat{A} b + c_2 \hat{A}^2 b + \cdots + c_k \hat{A}^k b = c_1 \hat{A} b|_{\mathcal{R}(\hat{A})} + c_2 \hat{A}^2 b|_{\mathcal{R}(\hat{A})} + \cdots + c_k \hat{A}^k b|_{\mathcal{R}(\hat{A})} = 0.$$

But, since

$$\begin{aligned} &\dim(\mathrm{span}\{\hat{A} b|_{\mathcal{R}(\hat{A})}, \hat{A}^2 b|_{\mathcal{R}(\hat{A})}, \cdots, \hat{A}^k b|_{\mathcal{R}(\hat{A})}\}) \\ &\quad = \dim(\hat{A}\,\mathrm{span}\{b|_{\mathcal{R}(\hat{A})}, \hat{A} b|_{\mathcal{R}(\hat{A})} \cdots, \hat{A}^{k-1} b|_{\mathcal{R}(\hat{A})}\}) = \dim(\hat{A}\mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})})) = k \end{aligned}$$

holds from Lemma 3.5, we have $c_1 = c_2 = \cdots = c_k = 0$, which implies $\dim(\mathcal{K}_{k+1}(\hat{A}, b)) = k + 1$. □

Corollary 3.8. *Assume* $\mathcal{N}(\hat{A}) = \mathcal{N}(\hat{A}^\mathsf{T})$, $\mathrm{grade}(\hat{A}, b|_{\mathcal{R}(\hat{A})}) = k$, *and* $b \notin \mathcal{R}(\hat{A})$. *Then,* $\dim(\mathcal{K}_{k+1}(\hat{A}, b)) = k + 1$ *holds.*

Proof. $\mathcal{N}(\hat{A}) = \mathcal{N}(\hat{A}^\mathsf{T})$ implies $\mathcal{N}(\hat{A}) \cap \mathcal{R}(\hat{A}) = \{0\}$. Hence, the corollary follows from Lemma 3.7. □

From Lemma 3.5, we have

$$\begin{aligned}\dim(\mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})})) &= \dim(\mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})})) \\ &= \dim(\hat{A}\mathcal{K}_k(\hat{A}, b|_{\mathcal{R}(\hat{A})})) \\ &= \dim(\hat{A}\mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})})) \\ &= k.\end{aligned}$$

Since $\mathcal{N}(\hat{A}) = \mathcal{N}(\hat{A}^\mathsf{T})$, we obtain $\hat{A}b|_{\mathcal{R}(\hat{A})} = \hat{A}b$ and $\dim(\hat{A}\mathcal{K}_{k+1}(\hat{A}, b)) = \dim(\hat{A}\mathcal{K}_{k+1}(\hat{A}, b|_{\mathcal{R}(\hat{A})})) = k$. If $b \notin \mathcal{R}(\hat{A})$ and $\dim(\hat{A}\mathcal{K}_k(\hat{A}, b)) = k$, $\dim(\mathcal{K}_{k+1}(\hat{A}, b)) = k + 1$ (See Lemma 3.7).

Let x_0 be the initial solution and $r_0 = b - \hat{A}x_0$. In the inconsistent case, a least squares solution is obtained at iteration k, and at iteration $k + 1$ breakdown occurs because of $\dim(\hat{A}\mathcal{K}_{k+1}(\hat{A}, r_0)) < \dim(\mathcal{K}_{k+1}(\hat{A}, r_0))$, i.e. rank deficiency of $\min_{z \in \mathcal{K}_{k+1}(\hat{A}, r_0)} \|b - \hat{A}(x_0 + z)\|_2 = \min_{z \in \mathcal{K}_{k+1}(\hat{A}, r_0)} \|r_0 - \hat{A}z\|_2$[19]. This case is also called the benign breakdown[21].

However, even if $\mathcal{N}(\hat{A}) = \mathcal{N}(\hat{A}^\mathsf{T})$, when (3.13) is inconsistent, the least squares problem $\min_{z \in \mathcal{K}_k(\hat{A}, r_0)} \|r_0 - \hat{A}z\|_2$ may become ill-conditioned as shown below.

Brown and Walker [19] introduced an effective condition number to explain why GMRES fails to converge for inconsistent least squares problems

$$\min_{x \in \mathbb{R}^m} \|b - \hat{A}x\|_2, \tag{3.13}$$

where $\hat{A} \in \mathbb{R}^{m \times m}$ is singular, in the following Theorem 3.9.

Theorem 3.9. [19] *Assume $\mathcal{N}(\hat{A}) = \mathcal{N}(\hat{A}^\mathsf{T})$, and denote the least squares residual of (3.13) by r^*, the residual at the $(k-1)$st iteration by r_{k-1}. If $r_{k-1} \neq r^*$, then*

$$\kappa_2(A_k) \geq \frac{\|A_k\|_2}{\|\bar{A}_k\|_2} \frac{\|r_{k-1}\|_2}{\sqrt{\|r_{k-1}\|_2^2 - \|r^*\|_2^2}}, \tag{3.14}$$

where $A_k \equiv \hat{A}|_{\mathcal{K}_k(A,r_0)}$ and $\bar{A}_k \equiv \hat{A}|_{\mathcal{K}_k(A,r_0)+\mathrm{span}\{r^*\}}$. Here, $\hat{A}|_S$ is the restriction of $\hat{A}$ to a subspace $S \subseteq \mathbb{R}^m$.

Theorem 3.9 implies that GMRES suffers ill-conditioning for $b \notin \mathcal{R}(\hat{A})$ as $\|r_k\|$ approaches $\|r^*\|$. We can apply Theorem 3.9 to AB-GMRES for least-squares problems by setting $\hat{A} \equiv AA^\mathsf{T}$. Theorem 3.9 also implies that even if we choose B as A^T, which satisfies the conditions in Theorem 3.4, AB-GMRES still may not converge numerically because of the ill-conditioning of R_k, losing accuracy in the solution computed in finite-precision arithmetic when r_{k-1} approaches r^*.

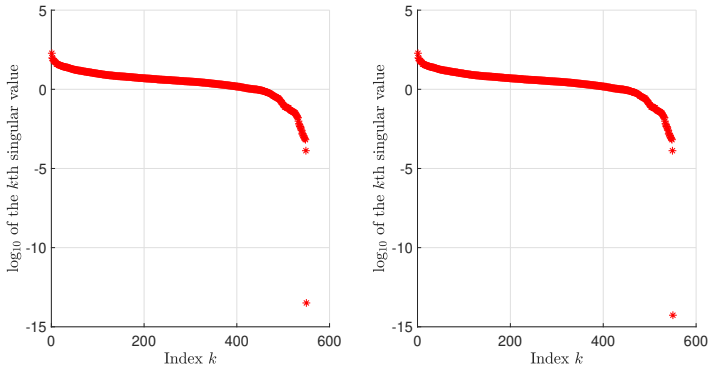
3.3 Deterioration of convergence of AB-GMRES applied to inconsistent underdetermined least squares problems

In this section, we illustrate, the deterioration of convergence of GMRES by numerical experiments. There are two points to note in this section. The first point is that the condition number of R_k tends to become very large as the iteration proceeds for inconsistent problems, as already mentioned in section 3.2.2. Due to $H_{k+1,k} = Q_{k+1}R_{k+1,k}$, the condition number of $H_{k+1,k}$ is the same as that of R_k, and will also become very large. The second point is as follows. Since $y_k = R_k^{-1}t_k$, y_k is obtained by applying backward substitution to the triangular system

$$R_k y_k = t_k. \tag{3.15}$$

When the triangular system becomes ill-conditioned, backward substitution becomes numerically unstable, and fails to give an accurate solution y_k.

Figure 3.1 shows that at step 550 the relative residual norm suddenly increases. To understand this increase, observe the singular values of R_{550}.



(a)Double precision arithmetic (b)Quadruple precision arithmetic

FIGURE 3.2: Singular value distribution of R_{550} for Maragal_3T in double and quadruple precision arithmetic.

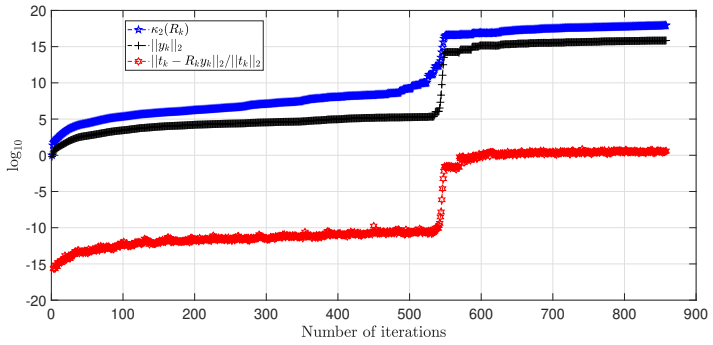


FIGURE 3.3: $\kappa_2(R_k)$, $\|y_k\|_2$, and $\|t_k - R_k y_k\|_2/\|t_k\|_2$ versus the number of iterations for Maragal_3T.

The left of Figure 3.2 shows the singular values of R_{550} which were computed in double precision arithmetic. The smallest singular value of R_{550} is 3.21×10^{-14}, which means that the triangular matrix R_{550} is very ill-conditioned and nearly singular in double precision arithmetic.

The right of Figure 3.2 shows the singular values of R_{550} which were computed in quadruple precision arithmetic using the Multiprecision Computing Toolbox for MATLAB [11]. The smallest singular value of R_{550} is 5.39×10^{-15}. Since quadruple precision is more accurate, from now on, we mainly show singular value distributions computed in quadruple precision.

Figure 3.3 shows $\kappa_2(R_k)$, $\|y_k\|_2$, and the relative residual norm $\|t_k - R_k y_k\|_2/\|t_k\|_2$ versus the number of iterations for AB-GMRES. The relative residual norm increases only gradually when the condition number of R_k is less than 10^8. When the condition number of R_k becomes larger than 10^{10}, the relative residual

norm starts to increase sharply. This observation shows that when the condition number of R_k becomes very large, the backward substitution will fail to give an accurate y_k. As a result, we would not get an accurate x_k, and the convergence of AB-GMRES would deteriorate.

3.4 Stabilized GMRES method

In this section, we first propose and present a stabilized GMRES method. Then, we explain its regularization effect comparing it with other regularization techniques.

3.4.1 The stabilized GMRES

In order to overcome the deterioration of convergence of GMRES for inconsistent systems, we propose solving the normal equations

$$R_k^{\mathsf{T}} R_k y_k = R_k^{\mathsf{T}} t_k \tag{3.16}$$

instead of $R_k y_k = t_k$ of (3.15), which we will call the stabilized GMRES. We replace line 8 of Algorithm 4 by Algorithm 5. This makes the system consistent, and stabilizes the process, as will be shown in the following.

One may also consider using the normal equations of $H_{k+1,k}$. However, before breakdown, we use the standard AB-GMRES, which means we do not have to store $H_{k+1,k}$. We only store R_k and update it in each iteration, which is cheaper.

Figure 3.4 shows the relative residual norm $\|A^{\mathsf{T}} r_k\|_2 / \|A^{\mathsf{T}} r_0\|_2$ versus the number of iterations for the standard AB-GMRES and stabilized AB-GMRES with $B = A^{\mathsf{T}}$ for Maragal_3T. The stabilized method reaches the relative residual norm level of 10^{-11} which improves a lot compared to the standard method. The method which we used for solving the normal equations (3.16) is the standard Cholesky decomposition without pivoting.

This seems paradoxical, since forming the normal equations whose coefficient matrix $R_k^{\mathsf{T}} R_k$ would square the condition number compared to R_k, which would make the ill-conditioned problem even worse. Why can the stabilized AB-GMRES give a more

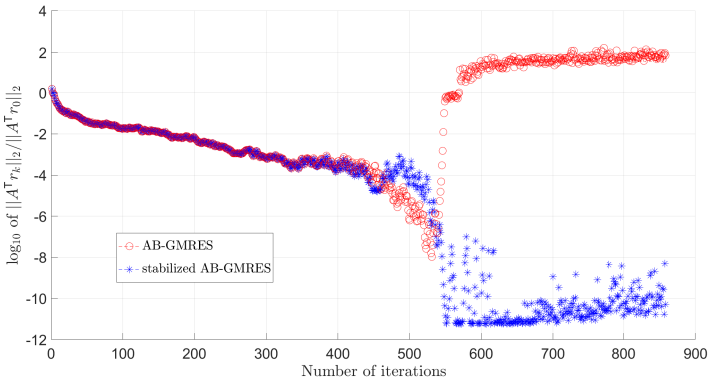


FIGURE 3.4: Comparison of the standard AB-GMRES with stabilized AB-GMRES for Maragal_3T.

Algorithm 5 Normal equations stabilization approach

1: Compute the QR decomposition of $H_{k+1,k} = Q_{k+1}R_{k+1,k}$.
2: $R_{k+1,k} = \begin{pmatrix} R_k \\ 0^{\mathsf{T}} \end{pmatrix}$, $\quad Q_{k+1}^{\mathsf{T}}\beta e_1 = \begin{pmatrix} t_k \\ \rho_{k+1} \end{pmatrix}$, $\quad \widetilde{R}_k = R_k^{\mathsf{T}}R_k$, $\quad \widetilde{t}_k = R_k^{\mathsf{T}}t_k$.
3: Compute the Cholesky decomposition of $\widetilde{R}_k = LL^{\mathsf{T}}$.
4: Solve $Lz_k = \widetilde{t}_k$ by forward substitution.
5: Solve $L^{\mathsf{T}}y_k = z_k$ by backward substitution.

accurate solution? We will explain why the stabilized AB-GMRES works in the next subsection.

In spite of the above mentioned merits of stabilization, solving the normal equations in AB-GMRES is expensive. Actually, we only need the stabilized AB-GMRES when R_k becomes ill-conditioned. Thus, we can speed up the process by switching AB-GMRES to stabilized AB-GMRES only when R_k becomes ill-conditioned. The condition number of an incrementaly enlarging triangular matrix can be estimated by techniques in [27]. In this paper, we adopt the switching strategy by monitoring the relative residual norm $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2$. Let $\mathrm{ATR}(k)$=$\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2$ for the kth iteration. When $\mathrm{ATR}(v)/\min_{k=1,2,\ldots,v-1}\mathrm{ATR}(k) > 10$, we judge that a jump in relative residual norm has occured, and we switch AB-GMRES to stabilized AB-GMRES at the vth iteration.

3.4.2 Why the stabilized GMRES method works

Consider solving $R_k y_k = t_k, R_k \in \mathbb{R}^{k\times k}, t_k \in \mathbb{R}^k$ by solving the normal equations (3.16), which, in theory, squares the condition number and makes the problem become

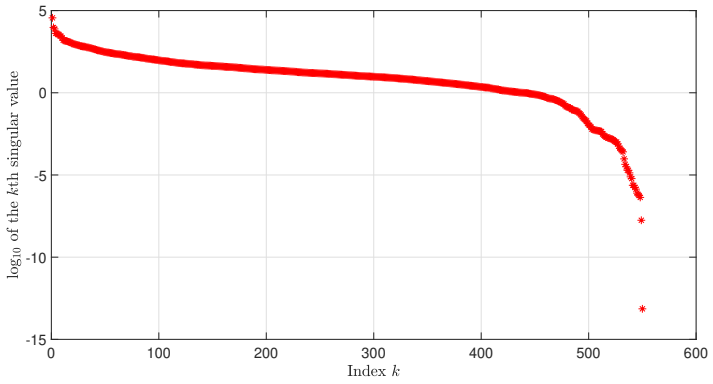


FIGURE 3.5: Singular values $\sigma_k(\mathrm{fl}_d(R_{550}^{\mathsf{T}}R_{550}))$, $k = 1, 2, \ldots, 550$ in quadruple precision arithmetic.

harder to solve numerically. However, in finite precision arithmetic, the condition number of the normal equations is not neccessarily squared. We will continue to illustrate the phenomenon by using the example in Section 3.3.

We used the MATLAB function `svd` in quadruple precision arithmetic [11] to calculate the singular values. The smallest singular value of R_{550} is 5.39×10^{-15}, so its square is 2.91×10^{-29}.

Let $\mathrm{fl}(\cdot)$ denote the evaluation of an expression in floating point arithmetic and $\mathrm{fl}_d(\cdot)$ and $\mathrm{fl}_q(\cdot)$ denote the result in double precision arithmetic and quadruple precision arithimetic, respectively. Figure 3.5 shows that, numerically, the smallest singular value of $\mathrm{fl}_d(R_{550}^{\mathsf{T}}R_{550})$ is 7.21×10^{-14}, which is much larger than 2.91×10^{-29}. Further, the Cholesky factor L of $\mathrm{fl}_d(R_{550}^{\mathsf{T}}R_{550}) = LL^{\mathsf{T}}$ computed in double precision precision arithmetic has the smallest singular value 3.50×10^{-7}, which is also larger than $\sqrt{2.91 \times 10^{-29}} = 5.39 \times 10^{-15}$. Thus, the triangular systems $Lz_k = \widetilde{t}_k$ and $L^{\mathsf{T}}y_k = z_k$ are better-conditioned than $R_k y_k = t_k$, which will ensure the stability of the forward and backward substitutions and succeeds in obtaining a much more accurate solution with stability compared to the standard approach as shown in Figure 3.4.

The left of Figure 3.6 compares the singular values $\sigma_k(\mathrm{fl}_d(R_{550}^{\mathsf{T}}R_{550}))$ and $\sigma_k(R_{550})^2, k = 1, 2, \ldots, 550$. The first to the 549th singular values of $\mathrm{fl}_d(R_{550}^{\mathsf{T}}R_{550})$ and the corresponding $\sigma(R_{550})^2$ are almost the same, while the last one is different. What will happen when R_k contains a cluster of small singular values?

The upper triangular matrix R_{610} contains a cluster of small singular values. The right of Figure 3.6 compares the singular values $\sigma_k(\mathrm{fl}_d(R_{610}^{\mathsf{T}}R_{610}))$ and $\sigma_k(R_{610})^2$. The

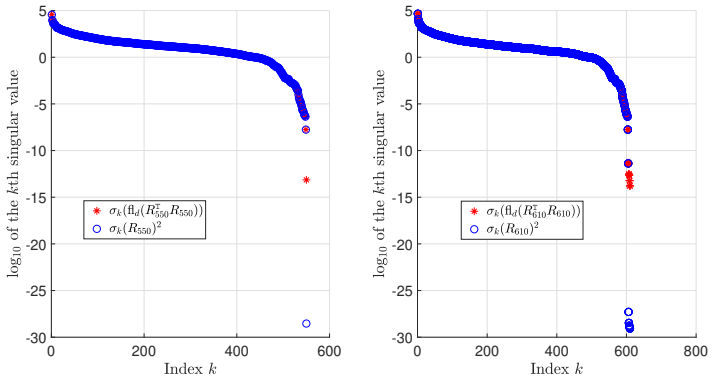


FIGURE 3.6: Singular values $\sigma_k(\mathrm{fl}_d(R_{550}^{\mathsf{T}}R_{550}))$, $\sigma_k(R_{550})^2$, $\sigma_k(\mathrm{fl}_d(R_{610}^{\mathsf{T}}R_{610}))$, and $\sigma_k(R_{610})^2$ in quadruple precision arithmetic.

larger singular values are the same as the 'exact' values, while the smaller singular values become larger than the 'exact' ones.

Experiment results show that finite precision arithmetic has the effect of shifting the tiny singular values upwards and reduce the condition number of $R^{\mathsf{T}}R$. Besides the fact that the possibly inconsistent system (3.15) is replaced by the consistent system (3.16), that is the reason why the normal equations (3.16) help to make the problem easier to solve.

Next, we computed $R_{550}^{\mathsf{T}}R_{550}$ in quadruple precision arithmetic and observed that the smallest singular values of $R_{550}^{\mathsf{T}}R_{550}$ coincided with the squared singular values $\sigma_k(R_{550})^2$ (blue circle symbol) in the left of Figure 3.6, unlike in double precision computation. Since the maximum of the elements of $|\mathrm{fl}_q(R_{550}^{\mathsf{T}}R_{550}) - \mathrm{fl}_d(R_{550}^{\mathsf{T}}R_{550})|$ is approximately 8.16×10^{-12}, double precision arithmetic contains error of the order of 10^{-12}. Thus, double precision arithmetic has an effect of regularizing the matrix $R_{550}^{\mathsf{T}}R_{550}$, since double precision matrix multiplication is not accurate enough to keep all the information.

3.4.3 Quadruple precision

In order to see the effect of the machine precision on the convergence of AB-GMRES, we compared the stabilized AB-GMRES with the standard AB-GMRES in quadruple precision arithmetic for the problem Maragal_3T in Figure 3.13 in terms of the relative residual norm $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}b\|_2$ versus the number of iterations. For both methods, the

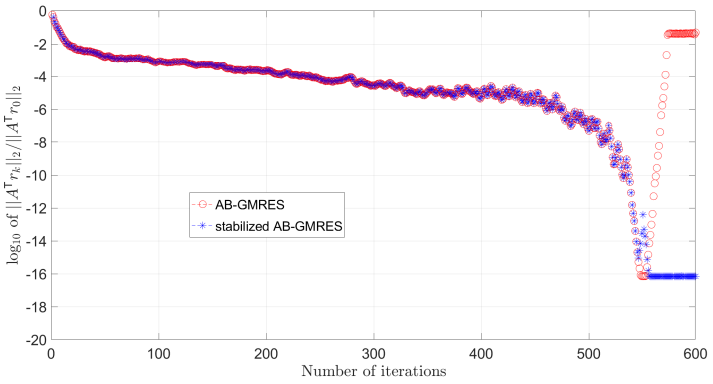


FIGURE 3.7: Effect of the stabilized method in quadruple precision arithmetic for Maragal_3T.

relative residual norm reached a lower level of order 10^{-16} compared to 10^{-12} and 10^{-8}, respectively, for double precision arithmetic in Figure 3.4. The curves of the relative residual norm became smoother compared to double precision. As seen in Figure 3.13, the relative residual norm of the standard AB-GMRES jumped to 10^{-1} after reaching 10^{-16}, whereas the relative residual norm of the stabilized GMRES stayed around 10^{-16}.

3.4.4 Rounding error analysis of the stabilized GMRES method

In order to understand the stability of the proposed method, we perform rounding error analysis. Let u be the unit roundoff [24], which is about 1.11×10^{-16} for the IEEE 754 binary 64 (double) in our experiments. The analysis shows that if the condition number of $R \in \mathbb{R}^{n\times n}$ is $\dfrac{1}{o(n\sqrt{u})}$, then the condition number of $R^{\mathsf{T}}R$ can be reduced from $\dfrac{1}{o(n^2u)}$ in exact arithmetic to $O\left(\dfrac{1}{n^2u}\right)$ in finite precision arithmetic.

Let $\mathrm{fl}(x)$ denote the floating point number corresponding to x. Let $A = (a_{ij}) \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$, and $|A| = (|a_{ij}|)$.

Then, we have (cf. [24])

$$|\mathrm{fl}(AB) - AB| \leq \gamma_n |A||B|, \quad \text{where} \quad \gamma_n := \frac{nu}{1-nu}. \tag{3.17}$$

Hence,

$$|\mathrm{fl}(R^{\mathsf{T}}R) - R^{\mathsf{T}}R| \leq \gamma_n |R^{\mathsf{T}}||R|. \tag{3.18}$$

Let

$$E = (\varepsilon_{ij}) := \mathrm{fl}(R^{\mathsf{T}}R) - R^{\mathsf{T}}R. \tag{3.19}$$

Then, we have

$$|E| = (|\varepsilon_{ij}|) \leq \gamma_n |R^{\mathsf{T}}||R|. \tag{3.20}$$

Let $R = U\Sigma V^{\mathsf{T}}$ be the singular value decomposition (SVD) of $R \in \mathbb{R}^{n\times n}$, where $U = [u_1, u_2, \ldots, u_n]$, $V = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

Then, note the following.

Lemma 3.10. $(|R^{\mathsf{T}}||R|)_{ij} \leq \|R\|_2^2 = \sigma_1^2$.

Proof. Let $R = [r_1, r_2, \ldots, r_n]$. Then,

$$\begin{aligned}
(|R^{\mathsf{T}}||R|)_{ij} &= |r_i|^{\mathsf{T}}|r_j| = (|r_i|, |r_j|) \\
&\leq \|r_i\|_2 \|r_j\|_2 \leq \max_{1\leq i\leq n} \|r_i\|_2^2 \\
&= \max_{i=1,\ldots,n} \|Re_i\|_2^2 \leq \max_{\|x\|_2=1} \|Rx\|_2^2 \\
&\leq \|R\|_2^2 = {\sigma_1}^2.
\end{aligned}$$

Here, e_i is the ith column of the identity matrix. □

Hence, we have

$$|\varepsilon_{ij}| \leq \gamma_n (|R^{\mathsf{T}}||R|)_{ij} \leq \gamma_n {\sigma_1}^2. \tag{3.21}$$

Let A, B, C be $n \times n$ Hermitian matrices, and $A = B + C$. Denote the eigenvalues of A, B and C by $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$, $\lambda_1(B) \geq \lambda_2(B) \geq \cdots \geq \lambda_n(B)$, and $\lambda_1(C) \geq \lambda_2(C) \geq \cdots \geq \lambda_n(C)$, respectively. Then, the following Weyl's inequality (See e.g. [28, 29])

$$\lambda_k(B) + \lambda_n(C) \leq \lambda_k(A) \leq \lambda_k(B) + \lambda_1(C), \qquad k = 1, 2 \ldots, n, \tag{3.22}$$

holds.

Hence, we have

$$|\lambda_k(A) - \lambda_k(B)| \leq \|C\|_2 = \|A - B\|_2, \qquad k = 1, 2, \ldots, n. \tag{3.23}$$

Letting $A = \text{fl}(R^\mathsf{T}R), B = R^\mathsf{T}R, C = E$, we have

$$|\lambda_k(\text{fl}(R^\mathsf{T}R)) - \lambda_k(R^\mathsf{T}R)| \leq \|E\|_2, \qquad k = 1, 2, \ldots, n. \tag{3.24}$$

Let $\text{fl}(R^\mathsf{T}R) = \widetilde{V}\widetilde{\Sigma}^2\widetilde{V}^\mathsf{T}$ be the SVD of $\text{fl}(R^\mathsf{T}R)$, where $\widetilde{\Sigma} = \text{diag}(\widetilde{\sigma}_1, \widetilde{\sigma}_2, \ldots, \widetilde{\sigma}_n)$, $\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \cdots \geq \widetilde{\sigma}_n \geq 0$, which gives

$$|\widetilde{\sigma}_k^2 - \sigma_k^2| \leq \|E\|_2, \qquad k = 1, 2, \ldots, n. \tag{3.25}$$

Note

$$\begin{aligned} |\widetilde{\sigma}_k^2 - \sigma_k^2| &\leq \|E\|_2 \leq \|E\|_F \\ &= \sqrt{\sum_{i,j=1}^{n} |\varepsilon_{ij}|^2} \leq \sqrt{\sum_{i,j=1}^{n} (\gamma_n {\sigma_1}^2)^2} \\ &= \sqrt{n^2(\gamma_n {\sigma_1}^2)^2} = n\gamma_n\sigma_1^2, \qquad k = 1, 2, \ldots, n. \end{aligned} \tag{3.26}$$

Hence,

$$|\widetilde{\sigma}_k^2 - \sigma_k^2| \leq n\gamma_n\sigma_1^2, \qquad k = 1, 2, \ldots, n, \tag{3.27}$$

i.e

$$\sigma_k^2 - n\gamma_n\sigma_1^2 \leq \widetilde{\sigma}_k^2 \leq \sigma_k^2 + n\gamma_n\sigma_1^2, \qquad k = 1, 2, \ldots, n, \tag{3.28}$$

or

$$\widetilde{\sigma}_k^2 = \sigma_k^2 + t_k n\gamma_n\sigma_1^2, \quad -1 \leq t_k \leq 1, \qquad k = 1, 2, \ldots, n. \tag{3.29}$$

Recall $\gamma_n = \dfrac{nu}{1 - nu}$, $u \approx 1.11 \times 10^{-16}$. If $nu \ll 1 \iff n \ll \dfrac{1}{u}$ ($\approx 9.01 \times 10^{15}$ for double precision arithmetic). Then,

$$\frac{1}{1 - nu} \approx 1 + nu \Longrightarrow \gamma_n = \frac{nu}{1 - nu} \approx nu(1 + nu) \approx nu. \tag{3.30}$$

Hence,

$$\widetilde{\sigma}_k^2 \approx \sigma_k^2 + t_k n^2 u \sigma_1^2, \quad -1 \le t_k \le 1, \qquad k = 1, 2, \ldots, n. \tag{3.31}$$

Then,

$$\widetilde{\sigma}_1^2 \approx \sigma_1^2 (1 + t_1 n^2 u), \qquad -1 \le t_1 \le 1. \tag{3.32}$$

We define the following Landau's symbols:

$$f(x) = O(g(x)) \text{ as } x \to \text{a denotes that } \frac{f(x)}{g(x)} \text{ is bounded as } x \to a, \tag{3.33}$$

and

$$f(x) = o(g(x)) \text{ as } x \to a \text{ denotes that } \lim_{x \to a} \frac{f(x)}{g(x)} = 0. \tag{3.34}$$

In the following, for instance, $o(n\sqrt{u})$ is defined by letting $x = n\sqrt{u}, a = 0$.

Assume $n\sqrt{u} \ll 1$ ($\Longleftrightarrow n \ll \dfrac{1}{\sqrt{u}} \approx 9.49 \times 10^7$). Then, since, $\dfrac{1}{\sqrt{u}} \ll \dfrac{1}{u}$, we have $n \ll \dfrac{1}{\sqrt{u}} \ll \dfrac{1}{u}$. Thus, $nu \ll 1$.

Assume

$$\kappa = \kappa(R) = \frac{\sigma_1}{\sigma_n} = \frac{1}{o(n\sqrt{u})}$$

$$\Longleftrightarrow \kappa^2 = \frac{1}{o(n^2 u)} \quad \Longleftrightarrow \quad \frac{1}{\kappa^2} = o(n^2 u) \tag{3.35}$$

holds. Then,

$$\begin{aligned} \widetilde{\sigma}_n^2 &\approx \sigma_n^2 + t_n n^2 u \sigma_1^2 = \sigma_1^2 \left(\frac{\sigma_n^2}{\sigma_1^2} + t_n n^2 u \right) \\ &= \sigma_1^2 \left(\frac{1}{\kappa^2} + t_n n^2 u \right) \approx \sigma_1^2 \left[o(n^2 u) + t_n n^2 u \right]. \end{aligned} \tag{3.36}$$

Assume $|t_n| > o(1)$. (Note that if we assume t_n is randomly distributed in the interval $[-1, 1]$, then, generically, $o(1) < |t_n|$ holds.) Then, since $-1 \le t_n \le 1$, $o(1) < |t_n| = O(1)$ holds. Hence, $\widetilde{\sigma}_n^2 \approx \sigma_1^2 t_n n^2 u$. Since $\widetilde{\sigma}_n^2 \ge 0$, we have $t_n > 0$ and $\dfrac{1}{t_n} = O(1)$.

Hence, from (3.32), (3.36) and $\dfrac{1}{t_n} = O(1)$, we have

$$\widetilde{\kappa}^2 = \kappa(\mathrm{fl}(R^{\mathsf{T}} R)) = \frac{\widetilde{\sigma}_1^2}{\widetilde{\sigma}_n^2} \approx \frac{1}{t_n n^2 u} \approx O\left(\frac{1}{n^2 u} \right). \tag{3.37}$$

In summary, we have the following theorem.

Theorem 3.11. *Let u be the unit roundoff, and $R \in \mathbb{R}^{n\times n}$. If $n\sqrt{u} \ll 1$ and $\kappa(R) = \dfrac{\sigma_1(R)}{\sigma_n(R)} = \dfrac{1}{o(n\sqrt{u})}$, then, generically, $\sigma_n(\mathrm{fl}(R^{\mathsf{T}}R)) \approx \sigma_1(R)^2\, t_n\, n^2 u$, where $o(1) < t_n = O(1)$, and $\kappa(\mathrm{fl}(R^{\mathsf{T}}R)) = O\left(\dfrac{1}{n^2 u}\right)$ hold.*

Remark 3.12. For IEEE double $u \approx 1.11 \times 10^{-16}$, $n\sqrt{u} \ll 1 \iff n \ll 9.49 \times 10^7$.

Remark 3.13. $\kappa(R^{\mathsf{T}}R) = \dfrac{1}{o(n^2 u)}$.

Then, numerical experiments suggest that if LL^{T} is the Cholesky decomposition of $\mathrm{fl}(R^{\mathsf{T}}R)$ computed in finite precision, then $\kappa(L) = O\left(\dfrac{1}{n\sqrt{u}}\right)$, even when $\kappa(R) = \dfrac{1}{o(n\sqrt{u})}$.

Thus, forming the normal equations and applying Cholesky decomposition can lead to a more stable computation for extremely ill-conditioned systems of equations, and hence explains why the stabilized GMRES method works without choosing the value of a regularization parameter such as in TSVD or Tikhonov regularization, which will be mentioned in § 3.5.1.1 and 3.5.1.2, respectively.

Let us compare estimates with numerical results for the Maragal_3T matrix in Figure 3.6.

For R_{550}, $n = 550$, $\sigma_1(R_{550}) \approx 1.90 \times 10^2$, $\sigma_{550}(R_{550}) \approx 5.39\times^{-15}$. Hence,

$$\kappa(R_{550}) = \frac{\sigma_1(R_{550})}{\sigma_{550}(R_{550})} \approx 3.53 \times 10^{16} = \frac{1}{o(n\sqrt{u})} \gg \frac{1}{n\sqrt{u}} \approx 1.73 \times 10^5.$$

Thus, $\widetilde{\sigma}_n^2 \approx \sigma_1^2\, t_n\, n^2 u \approx 1.21 \times 10^{-6}$, where $o(1) < t_n = O(1)$, and $\kappa(\mathrm{fl}(R^{\mathsf{T}}R)) \approx O\left(\dfrac{1}{n^2 u}\right) \approx 2.98 \times 10^{10}$, whereas in Figure 3.6, $\widetilde{\sigma}_n^2 \approx 7.21 \times 10^{-14}$, and $\kappa(\mathrm{fl}(R^{\mathsf{T}}R)) \approx 5.00 \times 10^{17}$.

For R_{610}, $n = 610$, $\sigma_1(R_{610}) \approx 2.13 \times 10^2$, $\sigma_{610}(R_{610}) \approx 2.91 \times 10^{-15}$. Hence,

$$\kappa(R_{610}) = \frac{\sigma_1(R_{610})}{\sigma_{610}(R_{610})} \approx 7.32 \times 10^{16} = \frac{1}{o(n\sqrt{u})} \gg \frac{1}{n\sqrt{u}} \approx 1.56 \times 10^5.$$

Thus, $\widetilde{\sigma}_n^2 \approx \sigma_1^2\, t_n\, n^2 u \approx 1.87 \times 10^{-6}$, where $o(1) < t_n = O(1)$, and $\kappa(\mathrm{fl}(R^{\mathsf{T}}R)) \approx O\left(\dfrac{1}{n^2 u}\right) \approx 2.42 \times 10^{10}$, whereas in Figure 3.6, $\widetilde{\sigma}_n^2 \approx 1.62 \times 10^{-14}$, and $\kappa(\mathrm{fl}(R^{\mathsf{T}}R)) \approx 2.77 \times 10^{18}$.

TABLE 3.2: Comparison of estimates and numerical experiments for Maragal_3T

	R_{550} $(n = 550)$		R_{610} $(n = 610)$	
	$\widetilde{\sigma}_n^2$	$\widetilde{\sigma}_1^2/\widetilde{\sigma}_n^2$	$\widetilde{\sigma}_n^2$	$\widetilde{\sigma}_1^2/\widetilde{\sigma}_n^2$
Estimates	1.21×10^{-6}	2.98×10^{10}	1.87×10^{-6}	2.42×10^{10}
Numerical experiment (Figure 3.6)	7.21×10^{-14}	5.00×10^{17}	1.62×10^{-14}	2.77×10^{18}
	σ_n^2	σ_1^2/σ_n^2	σ_n^2	σ_1^2/σ_n^2
	2.91×10^{-29}	1.25×10^{33}	8.47×10^{-30}	5.36×10^{33}

We summarize the results in Table 3.2. We think there are two reasons for the overestimation of $\tilde{\sigma}_n^2$. One comes from the inequality $\|E\|_2 \leq \|E\|_F$ in (3.26). The other is that $t_n > o(1)$, but t_n may be considerably samller than 1 in (3.29).

We remark that [30] analyzes the stability of the CholeskyQR2 algorithm using similar techniques. However, they assume $\kappa(R) \leq O\left(\frac{1}{\sqrt{u}}\right)$, whereas we assume $\kappa(R) = \frac{1}{o(n\sqrt{u})}$.

3.4.5 Two advantages of forming the normal equations

When R is singular, R^{-1} does not exist, and

$$Ry = t \tag{3.38}$$

does not have a solution when $t \notin \mathcal{R}(R)$.

If we reformulate (3.38) as a least squares problem

$$\min_y \|t - Ry\|_2, \tag{3.39}$$

then (3.39) has a solution even when $t \notin \mathcal{R}(R)$. For instance, the minimum-norm solution of (3.39) is given by $y = R^\dagger t$, where $R^\dagger$ is the pseudo-inverse of R.

Note that (3.39) is equivalent to the normal equations

$$R^\mathsf{T} Ry = R^\mathsf{T} t. \tag{3.40}$$

(3.40) is consistent, i.e. $R^\mathsf{T} t \in \mathcal{R}(R^\mathsf{T}) = \mathcal{R}(R^\mathsf{T} R)$, and has a solution.

Now consider the case when R is nearly singular (severely ill-conditioned), that is $\kappa(R) = \dfrac{1}{o(n\sqrt{u})}$. Then, solving (3.38) by backward substitution fails to give an accurate solution as shown in Figure 3.3.

On the other hand, we may expect that we may obtain a numerical solution of the least squares problem (3.39), for instance by approximating $y = R^{\dagger}t$ [31].

In fact, since (3.39) is equivalent to the normal equations, as we have seen in Theorem 3.11, $\kappa(\mathrm{fl}(R^{\mathsf{T}}R)) = O\left(\dfrac{1}{n^2u}\right)$ holds while $\kappa(R^{\mathsf{T}}R) = \dfrac{1}{o(n^2u)}$, which gives a numerical advantage.

Thus, we may say that forming the normal equations (3.40) has two advantages over the system of equations (3.38). One is that, it makes the system consistent and guarantees the existence of a solution, which opens the possibility of a numerical solution by some kind of approximation. The other advantage is that the normal equations become numerically better conditioned than in exact arithmetic.

These two effects work at the same time. Still, one may wonder which effect is dominant in improving the convergence of GMRES.

Consider executing AB-GMRES in double and quadruple precision arithmetic and compare t for the same iteration k. Denote R and t in double precision arithmetic as R_d and t_d and in quadruple precision arithmetic as R_q and t_q.

Solve $R_d\ y_{ds} = t_d$ by the stabilized AB-GMRES, which is equivalent to solving $R_d^{\mathsf{T}}R_d\ y_{ds} = R_d^{\mathsf{T}}t_d$, and create a right-hand side $t_s = R_d\ y_{ds}$. Then, use backward substitution to solve $R_d\ y_s = t_s$. The solution y_s is nearly the same as y_{ds}, as shown in Figure 3.8. The relative error is $\|y_s - y_{ds}\|_2/\|y_{ds}\|_2 = 1.87 \times 10^{-11}$. However, t_d and t_s are different, especially for the components with large index, as shown in Figure 3.10 (b), which suggests that if we have a consistent t_s, we can get a good solution y_s even with ill-conditioned R_d by backward substitution. This suggests that consistency is important.

In infinite precision $\|Ry - t\|$ and $\|Ax - b\|$ can be minimized at the same time. Since, we know that t_q makes the relative residuals $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2$ converge to a very low level by backward substitution before the relative residuals jump, as shown in Figure 3.13, we regard t_q as a good approximation of t in infinite precision. There are differences between t_d and t_q, as shown in Figure 3.10 (a) (also for R_d and R_q,

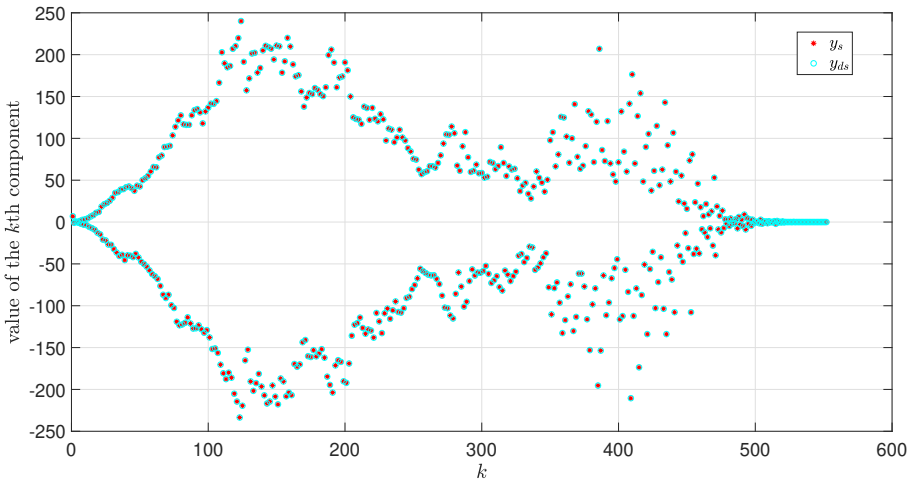


FIGURE 3.8: Comparison of y_s and y_{ds}.

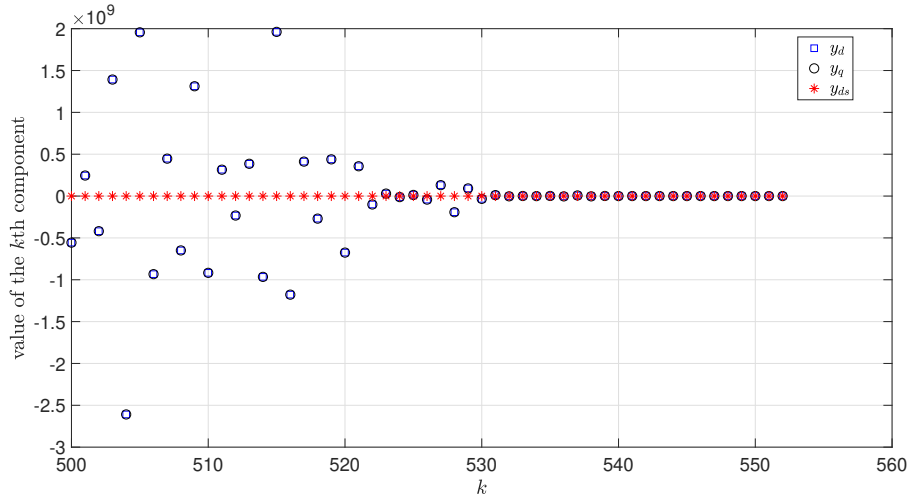


FIGURE 3.9: Comparison of y_d and y_q.

for example, the relative difference between the last component of Maragal3_T at 552 iteration $|R_d(552,552) - R_q(552,552)|/|R_q(552,552)| = 2.45 \times 10^{-1}$), especially for the components with large index, but t_s and t_q are similar, as shown in Figure 3.10 (c). Since, using t_q as right-hand side and performing back substitution by R_q can converge instead of t_d, and there are differences between them, we conclude that t_d contains noise.

Another point of view suggests using substitutions to solve $R_d^{\mathsf{T}} z = R_d^{\mathsf{T}} t_d$ first, then solve $R_d\, y = z$. This method fails to converge, since $z \notin \mathcal{R}(R_d)$. This also suggests that consistency is very important. Notice that, after forming the normal equations, and using Cholesky decomposition, we succeed to converge with L and L^{T} in Algorithm 5, which suggests that better conditioning makes $z_k \in \mathcal{R}(L^{\mathsf{T}})$. For example, for Maragal_3T

in 552 iterations, $\kappa(R_d) = 3.94 \times 10^{16}$, $\kappa(L) = 5.88 \times 10^{8}$. We guess that L does not have a null space numerically. This suggests that even if t_d contains noise, better conditioned L can reach a good solution y_{ds} rather than R_d, which shows better conditioning is also important.

We are sure that $t_d \notin \mathcal{R}(R_d)$, which suggests that t_d is inaccurate and contains noise. Thus, we tried to add noise to (3.16), and solve $R_i^{\mathsf{T}} R_i y = R_i^{\mathsf{T}} t + \eta$, where η is a random noise of magnitude $10^{-8}, 10^{-10}, 10^{-12}$, which is shown in Figure 3.12 (Note $\|\eta\|_2 = \|R_{552}^{\mathsf{T}}(t_s - t_q)\|_2 = 1.27 \times 10^{-7}$). The figure shows that the method still convergences with noise, but not as well as without noise. Better conditioning makes (3.16) insensitive to noise. This suggests better conditioning helps to resist the noises in t_d and makes the system stable.

3.4.5.1 Properties of the solution given by the stabilized method

When using GMRES to solve

$$\min \|b - Ax\|_2, \tag{3.41}$$

there is a mathematically equivalent problem

$$\min \|t_d - R_d y\|_2 \tag{3.42}$$

to solve for each GMRES iteration in double precision arithmetic. There are four ways to solve (3.42). They are backward substitution in double precision arithmetic (y_d), the normal equations approach in double precision arithmetic (y_{ds}), backward substitution in quadruple precision arithmetic (y_q), the normal equations approach in quadruple precision arithmetic (y_{qs}). The results are shown in the Table 3.3 and Table 3.4.

y_q reached the smallest relative residual and relative normal residual for (3.42), y_q is the least squares solution for (3.42). In terms of the relative error y_d is closer to y_q than y_{ds}. However, the most interesting thing is that y_{ds} gives the best least squares solution for (3.41), instead of y_q. This observation shows that successfully solving (3.42) cannot ensure solving (3.41) successfully. (3.41) and (3.42) are not equivalent due to noise in t in double precision arithmetic.

	y_d	y_{ds}	y_q	y_{qs}
$\frac{\|t_d - R_d y\|_2}{\|t_d\|_2}$	3.21×10^{-2}	7.55×10^{-2}	2.35×10^{-20}	3.27×10^{-4}
$\frac{\|R_d^{\mathsf{T}}(t_d - R_d y)\|_2}{\|R_d^{\mathsf{T}} t_d\|_2}$	5.08×10^{-1}	2.57×10^{-13}	9.21×10^{-20}	6.97×10^{-19}
$\frac{\|y - y_q\|_2}{\|y_q\|_2}$	4.86×10^{-13}	1.00	0	4.32×10^{-3}
$\frac{\|A^{\mathsf{T}}(b - Ax)\|_2}{\|A^{\mathsf{T}} b\|_2}$	4.56×10^{-2}	4.21×10^{-12}	4.92×10^{-2}	4.94×10^{-2}

TABLE 3.3: For Maragal_3T at iter=552.

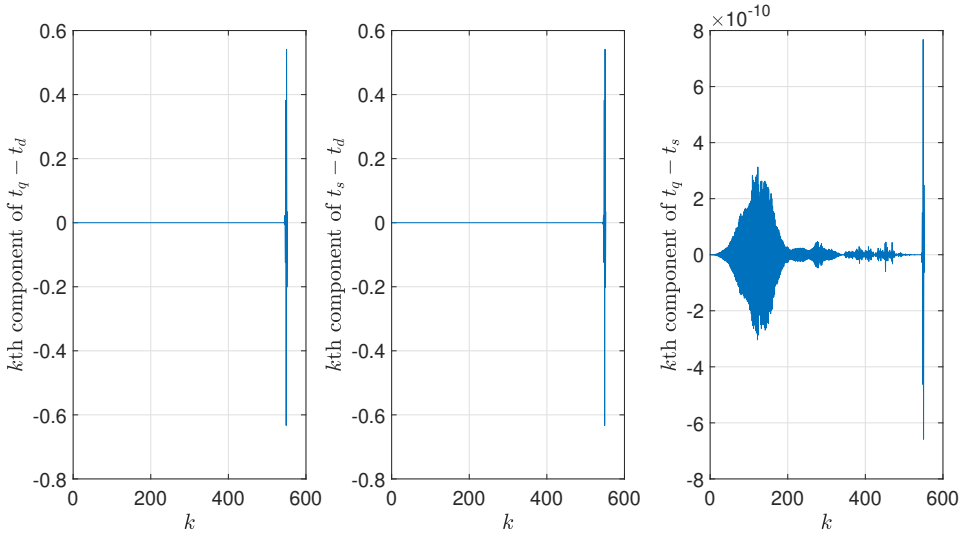
	y_d	y_{ds}	y_q	y_{qs}
$\frac{\|t_d - R_d y\|_2}{\|t_d\|_2}$	1.37	9.12×10^{-1}	1.19×10^{-18}	2.45×10^{-3}
$\frac{\|R^{\mathsf{T}}(t_d - R_d y)\|_2}{\|R_d^{\mathsf{T}} t_d\|_2}$	3.23×10^{1}	5.41×10^{-13}	2.29×10^{-17}	2.70×10^{-16}
$\frac{\|y - y_q\|_2}{\|y_q\|_2}$	1.49×10^{-14}	1.00	0	2.69×10^{-3}
$\frac{\|A^{\mathsf{T}}(b - Ax)\|_2}{\|A^{\mathsf{T}} b\|_2}$	2.97×10^{3}	3.14×10^{-11}	3.91×10^{2}	3.92×10^{2}

TABLE 3.4: For bw42 at iter=220.

When solving (3.41) in double precision arithmetic, the corresponding coefficient matrix and right-hand side of (3.42) are R_d and t_d, respectively. For quadruple precision arithmetic, we have R_q and t_q. For iteration 552, we get a solution x by using backward substitution for solving $R_q\ y^* = t_q$, and found the relative error for the same steps by the stabilized method x_{ds} is $\frac{\|x_{ds} - x_q\|_2}{\|x_q\|} = 7.81 \times 10^{-10}$. Thus, we confirmed that the stabilized method seems to converge to the least squares solution of (3.41). Notice that y^* and y_{ds} are not close ($\|y^* - y_{ds}\|_2/\|y^*\|_2 = 1.00$), since V_d and V_q are different, as shown in Figure 3.11. Note that, $x_{ds} = V_d\ y_{ds}$, $x_q = V_q\ y^*$. The column vectors of V_d lose orthogonality gradually, especially for the last several columns, where the inner product between different columns of V_d are of order one. Observe that in Figure 3.8 and Figure 3.9, the components of y_{ds} with large k ($k \geq 500$) are tiny (it decreases from 2.76 gradually to 1.75×10^{-10}), whereas y_d and y_q decrease from 1.96×10^{9} to 9.59×10^{-2}, which means the solution x_{ds} is less effected by the loss of orthogonality.

From Figure 3.10, regard t_q as accurate, then, $t_q - t_d$ is the total noise which t_d contains, $t_s - t_d$ is the inconsistent noise, $t_q - t_s$, whose magnitude is of order 10^{-10} is very tiny. From Figure 3.12, this is acceptable.

Concluding the above, the stabilized method can converge to the least squares solution



(a)total noise (b)inconsistent noise (c)'consistent' noise=(a)-(b)

FIGURE 3.10: The noise in t.

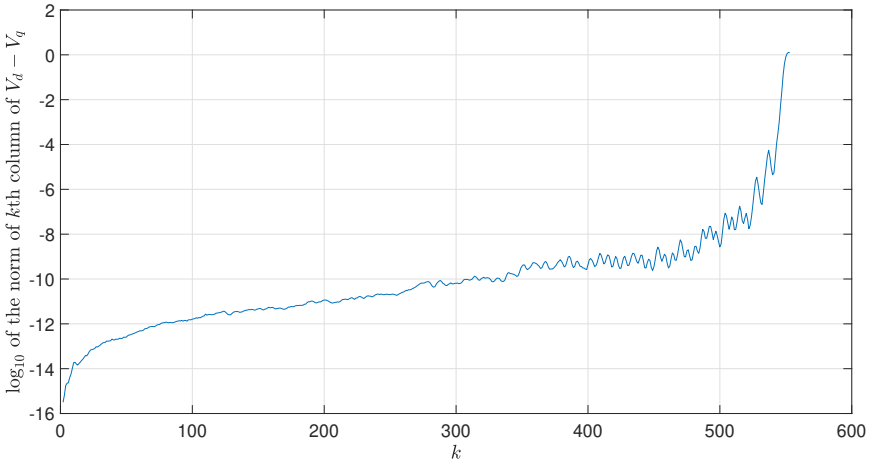


FIGURE 3.11: Norm of the columns of $V_d - V_q$.

even if t_d contains noise. Consistency eliminated the inconsistent noise in t_d. Better conditioning made the system insensitive to noise, as seen in Figure 3.10, and also made the generated L become better conditioned.

3.4.6 Quadruple precision

In order to see the effect of the machine precision ϵ on the convergence of the AB-GMRES, we compared the stabilized AB-GMRES with the AB-GMRES in quadruple precision arithmetic for the problem Maragal_3T in Figure 3.13. For both methods, the relative residual norm reached a smaller level of 10^{-16} compared to 10^{-12} and 10^{-8},

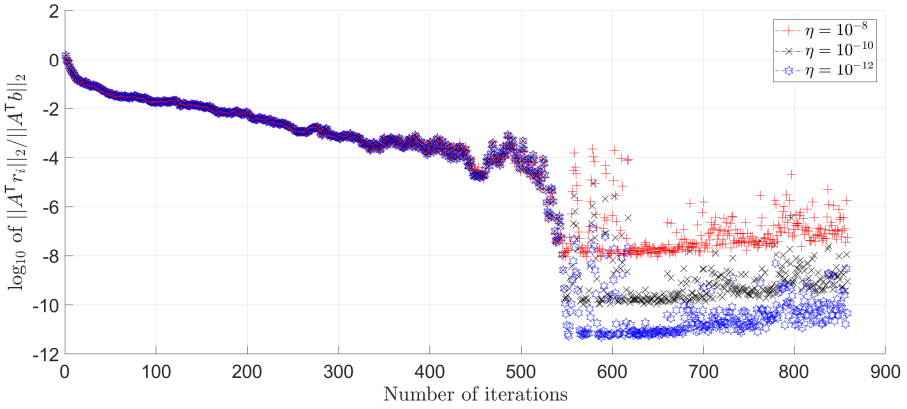


FIGURE 3.12: The stabilized method with noises for Maragal_3T.

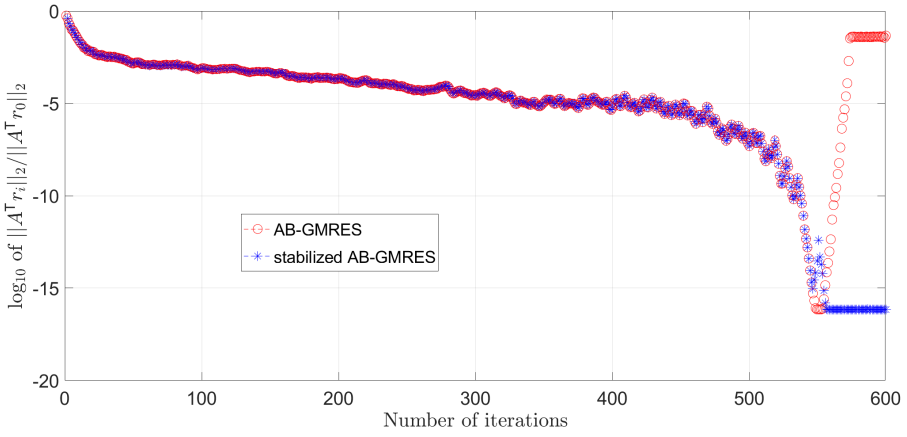


FIGURE 3.13: Effect of the stabilized method in quadruple precision arithmetic for Maragal_3T.

respectively, for double precision arithmetic in Figure 3.4. The curve of the relative residual norm became smoother compared to double precision. As seen in Figure 3.13, the relative residual norm of the AB-GMRES method jumped to 10^{-1} after reaching 10^{-16}, whereas the relative residual norm of the stabilized GMRES stayed around 10^{-16}.

3.4.7 When the stabilized GMRES method works

The stabilized GMRES does not always stabilize the solution of the upper triangular system. A counter example is when R_k is a Läuchli matrix [32], implying that $R_k^T R_k$ computed in finite precision becomes singular. Indeed, when GMRES is applied to a

linear system with an EP (equal preojecton) matrix A_3, that is $\mathcal{N}(A_3)$=$\mathcal{N}(A_3^{\mathsf{T}})$ such as

$$A_3 x = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} - \frac{\sqrt{6u}}{6} & -\frac{\sqrt{6u}}{6} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} + \frac{\sqrt{6u}}{6} & \frac{\sqrt{6u}}{6} \\ 0 & \frac{\sqrt{6u}}{3} & \frac{\sqrt{6u}}{3} \end{pmatrix} x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \tag{3.43}$$

where A_3 has the null space $\mathcal{N}(A_3) = \mathrm{span}\{(1,-1,1)^{\mathsf{T}}\}$, and u is the unit roundoff, the resulting R_k is a Läuchli matrix.

Apply GMRES with $x_0 = 0$ to (3.43). Let $R_k \in \mathbb{R}^{k\times k}$ be the upper triangular matrix obtained at the kth iteration of GMRES. In the second iteration, after applying the Givens rotation to $H_{3,2}$, we obtain the following:

$$R_2 = \begin{pmatrix} 1 & 1 \\ 0 & \sqrt{u} \end{pmatrix}, \qquad R_2^{\mathsf{T}} R_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1+u \end{pmatrix} \simeq \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \tag{3.44}$$

Thus, there is a risk that the stabilized GMRES will give a numerically singular matrix $R_2^{\mathsf{T}} R_2$ in finite precision arithmetic for nonsingular R_2. We will analyze this phenomenon.

Note that the following theorem holds from Theorem 8.10 of [24], where $|b| = (|b_1|, |b_2|, \ldots, |b_n|)^{\mathsf{T}}$ for $b = (b_1, b_2, \ldots, b_n)^{\mathsf{T}} \in \mathbb{R}^n$.

Theorem 3.14. *Let $T = (t_{ij}) \in \mathbb{R}^{n\times n}$ be a triangular matrix and $b \in \mathbb{R}^n$. Then, the computed solution $\hat{x}$ obtained from substitution applied to $Tx = b$ satisfies*

$$\hat{x} = x + O(n^2 u) M(T)^{-1} |b|. \tag{3.45}$$

Here, $M(T) = (m_{ij})$ is the comparison matrix such that

$$m_{ij} = \begin{cases} |t_{ij}|, & i = j, \\ -|t_{ij}|, & i \neq j. \end{cases} \tag{3.46}$$

Further, we define the following. Let

$$\mathbb{O}(x) = \begin{pmatrix} O(x) \\ O(x) \\ \vdots \\ O(x) \end{pmatrix} \in \mathbb{R}^n, \quad \mathcal{O}(x) = [\mathbb{O}(x), \mathbb{O}(x), \cdots, \mathbb{O}(x)] \in \mathbb{R}^{n\times n}. \tag{3.47}$$

We assume that the basic arithmetic operations op $= +, -, *, /$ satisfy $\mathrm{fl}(x \text{ op } y) = (x \text{ op } y)(1 + O(u))$ as in [24].

Let $x, y \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$. Then,

$$\mathrm{fl}(x^\mathsf{T} y) = x^\mathsf{T} y + O(nu)|x|^\mathsf{T}|y| = x^\mathsf{T} y + O(nu),$$

$$\mathrm{fl}(Ax) = Ax + \mathbb{O}(nu)|A||x| = Ax + \mathbb{O}(nu).$$

Let $C \in \mathbb{R}^{n\times n}$ and $\|C\|_2 = O(1)$. We say $C \in \mathbb{R}^{n\times n}$ is numerically nonsingular if the statement

$$\mathrm{fl}(Cx) = \mathbb{O}(u) \quad \Rightarrow \quad x = \mathbb{O}(u) \tag{3.48}$$

holds. Note that this definition of numerical nonsingularity agrees with that of numerical rank [18] due to the following.

Let the SVD of $C = U\Sigma V^\mathsf{T}$, where U, V are orthogonal matrices and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$. We assume $\|C\|_2 = \sigma_1 = O(1)$. If the numerical rank of C is $r < n$, there is a singular value $\sigma_i = O(u)$, $r + 1 \le i \le n$. Then, $Cx = U\Sigma V^\mathsf{T} x = \mathbb{O}(u)$ admits $x' = V^\mathsf{T} x = (x'_1, x'_2, \ldots, x'_n)^\mathsf{T}$ such that $x'_i = O(1)$, and hence $x = \mathbb{O}(1)$. Thus, C is numericaly singular. Then, the following theorem holds.

Theorem 3.15. *Let $R_k = (r_{ij}) \in \mathbb{R}^{k\times k}$ be an upper-triangular matrix and*

$$R_{k+1} = \begin{pmatrix} R_k & d \\ 0^\mathsf{T} & r_{k+1,k+1} \end{pmatrix} \in \mathbb{R}^{(k+1)\times(k+1)}. \tag{3.49}$$

Assume that R_k is nonsingular and numerically nonsingular, $R_k = \mathcal{O}(1)$, $R_k^{-1} = \mathcal{O}(1), M(R_k)^{-1} = \mathcal{O}(1), d = \mathbb{O}(1)$, and $O(k) = O(k^2) = O(1)$. Then, the following holds:

$$\mathrm{fl}(R_{k+1}^\mathsf{T} R_{k+1}) \text{ is numerically nonsingular } \iff \mathrm{fl}(r_{k+1,k+1}^2) > \mathrm{fl}(d^\mathsf{T} d)O(u).$$

The proof is as follows.

$$\mathrm{fl}(R_{k+1}^{\mathsf{T}}R_{k+1}) \textit{ is numerically nonsingular} \quad \Longleftrightarrow \quad \mathrm{fl}(r_{k+1,k+1}^2) > \mathrm{fl}(d^{\mathsf{T}}d)O(u).$$

Proof. Note that

$$R_{k+1}^{\mathsf{T}}R_{k+1} = \begin{pmatrix} R_k & 0 \\ d^{\mathsf{T}} & r_{k+1,k+1} \end{pmatrix} \begin{pmatrix} R_k & d \\ 0^{\mathsf{T}} & r_{k+1,k+1} \end{pmatrix} = \begin{pmatrix} R_k^{\mathsf{T}}R_k & R_k^{\mathsf{T}}d \\ d^{\mathsf{T}}R_k & d^{\mathsf{T}}d + r_{k+1,k+1}^2 \end{pmatrix}.$$

Proof of ($\Rightarrow$)

Assume $\mathrm{fl}(r_{k+1,k+1}^2) \le \mathrm{fl}(d^{\mathsf{T}}d)O(u)$. Then, since

$$\begin{aligned} \mathrm{fl}(d^{\mathsf{T}}d) &= d^{\mathsf{T}}d + O(ku)d^{\mathsf{T}}d = (1+O(ku))d^{\mathsf{T}}d, \\ \mathrm{fl}(d^{\mathsf{T}}d + r_{k+1,k+1}^2) &= (d^{\mathsf{T}}d + r_{k+1,k+1}^2)(1+O(ku)) = d^{\mathsf{T}}d(1+O(ku)), \\ R_k &= \mathcal{O}(1), \quad \text{and} \quad d = \mathbb{O}(1), \end{aligned}$$

we have

$$\begin{aligned} \mathrm{fl}(R_{k+1}^{\mathsf{T}}R_{k+1}) &= \begin{pmatrix} R_k^{\mathsf{T}}R_k + O(ku)|R_k|^{\mathsf{T}}|R_k| & R_k^{\mathsf{T}}d + O(ku)|R_k|^{\mathsf{T}}|d| \\ d^{\mathsf{T}}R_k + O(ku)|d|^{\mathsf{T}}|R_k| & d^{\mathsf{T}}d + O(ku)d^{\mathsf{T}}d \end{pmatrix} \\ &= \begin{pmatrix} R_k^{\mathsf{T}} \\ d^{\mathsf{T}} \end{pmatrix} \begin{pmatrix} R_k & d \end{pmatrix} + \mathcal{O}(ku). \end{aligned} \tag{3.50}$$

Note

$$\begin{pmatrix} R_k & d \end{pmatrix} \begin{pmatrix} -R_k^{-1}d \\ 1 \end{pmatrix} = -R_kR_k^{-1}d + d = 0,$$

since R_k is nonsingular.

Hence,

$$\mathrm{fl}(\begin{pmatrix} R_k & d \end{pmatrix} \begin{pmatrix} -R_k^{-1}d \\ 1 \end{pmatrix}) = \mathrm{fl}\{R_k\mathrm{fl}(-R_k^{-1}d) + d\} = [\mathrm{fl}\{R_k\mathrm{fl}(-R_k^{-1}d)\} + d]\{1+O(u)\}.$$

Note here that

$$fl\{R_k fl(-R_k^{-1}d)\} = R_k fl(-R_k^{-1}d) + O(ku)|R_k||R_k^{-1}d|,$$

and

$$fl(-R_k^{-1}d) = -R_k^{-1}d + O(k^2u)M(R_k)^{-1}|d| \tag{3.51}$$

from Theorem 3.14. Hence,

$$fl(\begin{pmatrix} R_k & d \end{pmatrix}\begin{pmatrix} -R_k^{-1}d \\ 1 \end{pmatrix}) = O(k^2u)R_sM(R_k)^{-1}|d| + O(ku)|R_k||R_k^{-1}d| = \mathbb{O}(k^2u),$$

since $R_k^{-1} = \mathcal{O}(1)$ and $M(R_k)^{-1} = \mathcal{O}(1)$.

Then,

$$fl(R_{k+1}^{\mathsf{T}}R_{k+1}\begin{pmatrix} -R_k^{-1}d \\ 1 \end{pmatrix})$$
$$= fl(\{\begin{pmatrix} R_k^{\mathsf{T}} \\ d^{\mathsf{T}} \end{pmatrix}\begin{pmatrix} R_k & d \end{pmatrix}+\mathcal{O}(ku)\}\begin{pmatrix} -R_k^{-1}d + \mathcal{O}(k^2u)M(R_k)^{-1}|d| \\ 1 \end{pmatrix}) = \mathbb{O}(k^2u) = \mathbb{O}(u),$$

since (3.50), (3.51), and $O(k^2) = O(1)$. Since $\begin{pmatrix} -R_k^{-1}d \\ 1 \end{pmatrix} = \mathbb{O}(1)$, $R_{k+1}^{\mathsf{T}}R_{k+1}$ is numerically singular. By contraposition, ($\Rightarrow$) holds.

Proof of ($\Leftarrow$)

Assume $R_{k+1}^{\mathsf{T}}R_{k+1}$ is not numerically nonsingular. Then, there exists a vector $\begin{pmatrix} z \\ w \end{pmatrix} \in$

$\mathbb{R}^{k+1}$ such that $\left|\begin{pmatrix} z \\ w \end{pmatrix}\right| > \mathbb{O}(u)$, and

$$\text{fl}\{R_{k+1}^{\mathsf{T}} R_{k+1} \begin{pmatrix} z \\ w \end{pmatrix}\} = R_{k+1}^{\mathsf{T}} \left(R_{k+1} \begin{pmatrix} z \\ w \end{pmatrix} + |R_{k+1}| \left|\begin{pmatrix} z \\ w \end{pmatrix}\right| O((k+1)u) \right) +$$
$$\left|R_{k+1}^{\mathsf{T}}\right| \left| R_{k+1} \begin{pmatrix} z \\ w \end{pmatrix} + |R_{k+1}| \left|\begin{pmatrix} z \\ w \end{pmatrix}\right| O((k+1)u) \right| O((k+1)u) = \mathbb{O}(u)$$

assuming $O(k+1) = O(1)$.

Hence,

$$\text{fl}\{R_{k+1}^{\mathsf{T}} R_{k+1} \begin{pmatrix} z \\ w \end{pmatrix}\} = \begin{pmatrix} R_k^{\mathsf{T}} R_k & R_k^{\mathsf{T}} d \\ d^{\mathsf{T}} R_k & d^{\mathsf{T}} d + r_{k+1,k+1}^2 \end{pmatrix} \begin{pmatrix} z \\ w \end{pmatrix} + \mathbb{O}(u) = \mathbb{O}(u).$$

Thus,

$$R_k^{\mathsf{T}} R_s z + w R_k^{\mathsf{T}} d = \mathbb{O}(u), \tag{3.52}$$

$$d^{\mathsf{T}} R_s z + (d^{\mathsf{T}} d + r_{k+1,k+1}^2) w = \mathbb{O}(u). \tag{3.53}$$

(3.52) can be expressed as $R_k^{\mathsf{T}}(R_s z + wd) = \mathbb{O}(u)$. From Lemma 3.16, R_k^{T} is numerically nonsingular, so that

$$R_s z + wd = \mathbb{O}(u). \tag{3.54}$$

Hence, from (3.53), $d^{\mathsf{T}} R_s z + w(d^{\mathsf{T}} d + r_{k+1,k+1}^2) = d^{\mathsf{T}}(R_s z + wd) + w r_{k+1,k+1}^2 = O(u)$. Thus, $w r_{k+1,k+1}^2 = O(u)$. If $w = O(u)$, $R_s z = \mathbb{O}(u)$ from (3.54). Since R_k is numerically nonsingular, $z = \mathbb{O}(u)$, which contradicts with the assumption.

Hence, $|w| > O(u)$, so that $r_{k+1,k+1}^2 = O(u)$, which gives

$$\text{fl}(r_{k+1,k+1}^2) = O(u) \leq \text{fl}(d^{\mathsf{T}} d) O(u).$$

□

Lemma 3.16. *Let $n = O(1)$. If $A \in \mathbb{R}^{n \times n}$ is numerically nonsingular, and $A^{-1} = \mathcal{O}(1)$, then A^{T} is numerically nonsingular.*

Proof. If

$$\text{fl}(A^{\mathsf{T}} x) = A^{\mathsf{T}} x + \mathbb{O}(nu)|A^{\mathsf{T}}||x| = \mathbb{O}(nu),$$

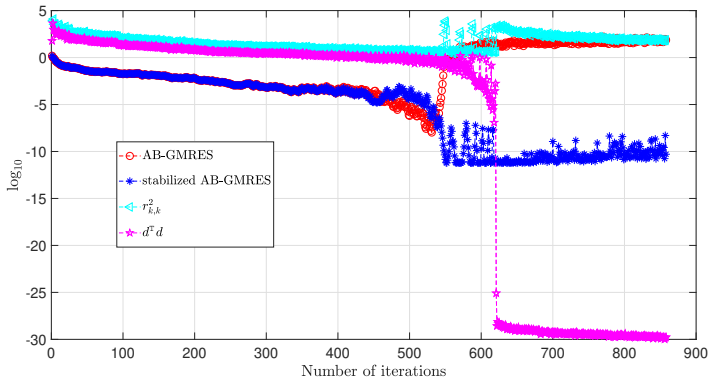


FIGURE 3.14: $r_{k,k}^2$, $d^\mathsf{T}d$, and $\|A^\mathsf{T}r_k\|_2/\|A^\mathsf{T}b\|_2$ for AB-GMRES and stabilized AB-GMRES for Maragal_3T.

then

$$\mathrm{fl}(x^\mathsf{T}A) = x^\mathsf{T}A + \mathbb{O}^\mathsf{T}(nu) = \mathbb{O}^\mathsf{T}(nu).$$

Thus,

$$\mathrm{fl}(x^\mathsf{T}Ay) = \mathrm{fl}(x^\mathsf{T}A)y + O(nu)|\mathrm{fl}(x^\mathsf{T}A)||y| = O(nu)$$

holds for all $y = \mathbb{O}(1)$.

For arbitrary $z = \mathbb{O}(1) \in \mathbb{R}^n$, let

$$y = A^{-1}z = \mathbb{O}(1).$$

Then,

$$\mathrm{fl}(Ay) = Ay + O(nu)|A||y| = z + O(nu)|A||y|.$$

Hence,

$$z = \mathrm{fl}(Ay) + O(nu)|A||y| = \mathrm{fl}(Ay) + \mathbb{O}(nu).$$

Thus, we have

$$\mathrm{fl}(x^\mathsf{T}z) = x^\mathsf{T}z + O(nu)|x|^\mathsf{T}|z| = \mathrm{fl}(x^\mathsf{T}Ay) + O(nu) = O(nu)$$

for arbitrary $z = \mathbb{O}(1) \in \mathbb{R}^n$. Hence, $x = \mathbb{O}(u)$, so that A^T is numerically nonsingular. □

Theorem 3.15 gives the necessary and sufficient condition so that the stabilized GMRES works at the $(k+1)$st iteration, i.e. $R_{k+1}^\mathsf{T}R_{k+1}$ is numerically nonsingular.

The difficulty in solving $R_i y_i = t_i$ by backward substitution is not necessarily because the diagonals of R_i are tiny. The reason is that R_i has tiny singular values. However, the exceptional example (3.44) exists where the stabilized AB-GMRES does not work. The condition $\mathrm{fl}(r_{k+1,k+1}^2) > \mathrm{fl}(d^\mathsf{T} d)O(u)$ in Theorem 3.15 excludes such exceptions.

Figure 3.14 shows $r_{k+1,k+1}^2$ and $d^\mathsf{T} d$ together with the relative residual norm $\|A^\mathsf{T} r_k\|_2 / \|A^\mathsf{T} b\|_2$ of AB-GMRES and stabilized AB-GMRES for Maragal_3T. The figure shows that up to 613 iterations, the conditions in Theorem 3.15 are satisfied, and $R_{k+1}^\mathsf{T} R_{k+1}$ is numerically nonsingular, so that the stabilized AB-GMRES works.

In fact, for

$$R_2 = \begin{pmatrix} 1 & 1 \\ 0 & \sqrt{u} \end{pmatrix}$$

of (3.44), $\sigma_1(R_2) \approx \sqrt{2}$, $\sigma_2(R_2) \approx \sqrt{\dfrac{u}{2}}$, so that $\kappa(R_2) \approx \dfrac{2}{\sqrt{u}} \approx O\left(\dfrac{1}{n\sqrt{u}}\right) \ll o\left(\dfrac{1}{n\sqrt{u}}\right)$, so the condition (3.4.4) is not satisfied, and the stabilized GMRES is not guaranteed to work in this case.

3.5 Comparisons with other methods

We show the numerical performance of the proposed stabilized AB-GMRES method on test matrices, compared with previous methods. All programs for iterative methods were coded according to the algorithms in [5, 6, 8, 33]. Each method was terminated at the iteration step which gives the minimum relative residual norm within m iterations, where m is the number of the rows of the matrix. No restarts were used for GMRES. Experiments were done for rank-deficient underdetermined matrices whose information is given in Table 1. Here, we have deleted the zero rows and columns of the test matrices beforehand. The elements of b were randomly generated using the MATLAB function `rand`. Therefore, generically $b \notin \mathcal{R}(A)$ and the problem is inconsistent. Each experiment was done 10 times for the same right-hand side b and the average of the CPU times are shown. The symbol - denotes that $\|A^\mathsf{T} r_k\|_2 / \|A^\mathsf{T} r_0\|_2$ did not reach 10^{-8} within m iterations. The symbol $(*)$ denotes that we used the MATLAB function `chol` instead of Cholesky decomposition without pivoting for solving the normal equations (3.16) to save CPU time, except for Havard500, for which Cholesky decomposition without pivoting did not converge. The symbol (&) denotes the case where even using the

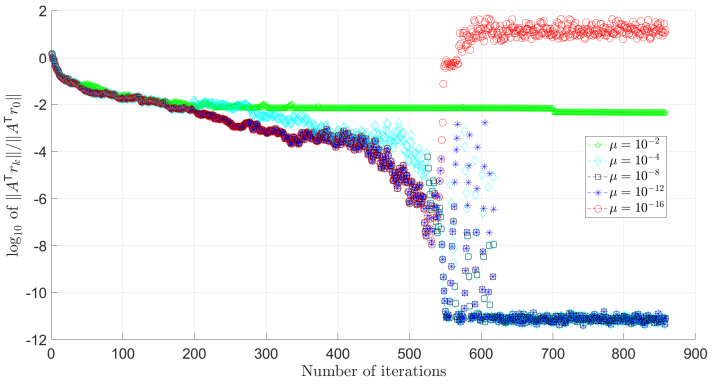


FIGURE 3.15: Relative residual norm for TSVD stabilized AB-GMRES versus number of iterations for different values of the regularization parameter μ for Maragal_3T.

MATLAB function `chol` for solving equation (3.16) failed to converge. Then, we used the MATLAB function backslash for solving the normal equations (3.16).

3.5.1 Underdetermined inconsistent least squares problems

3.5.1.1 Comparison with Truncated SVD method

Motivated by the stabilized AB-GMRES, we also applied the truncated singular value decomposition (TSVD) stabilization method and compared it with the stabilized AB-GMRES. The method modifies R_k by truncating singular values smaller than μ. More specifically, let $R_k = U\Sigma V^{\mathsf{T}}$ be the SVD of R_k, where the columns of $U = [u_1, u_2, \ldots, u_k]$ and $V = [v_1, v_2, \ldots, v_k]$ are the left and right singular vectors, respectively, and the diagonal entries of $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$ are the singular values of R_k in descending order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$. Then, the TSVD approximates $R_k \simeq \sum_{i=1}^{j} \sigma_i u_i v_i^{\mathsf{T}}$ with j such that $\sigma_{j+1} \leq \mu\sigma_1 \leq \sigma_j$ and $y_k = R_k^{-1} t_k \simeq \sum_{i=1}^{j} \frac{1}{\sigma_i} v_i u_i^{\mathsf{T}} t_i, j \leq k$.

For the problem Maragal_3T, when $\mu = 10^{-13}, 10^{-12}, \ldots, 10^{-4}$, the method converges but when μ is smaller than 10^{-13} or larger than 10^{-4}, it does not converge as shown in Figure 3.15. Numerical experiments showed that $\mu = \sqrt{u} \simeq 10^{-8}$, where u is the unit roundoff (about 10^{-16} in double presion arithmetic), gave the best result among $\mu = 10^{-1}, 10^{-2}, \ldots, 10^{-16}$ in terms of the relative residual norm. The convergence behaviour of the TSVD stabilization method with $\mu = 10^{-8}$ is similar to the stabilized AB-GMRES method as shown in Figure 3.16, which suggests that eliminating tiny singular values of R_k which are less than 10^{-8} is effctive for sovling problem (3.1). However, the

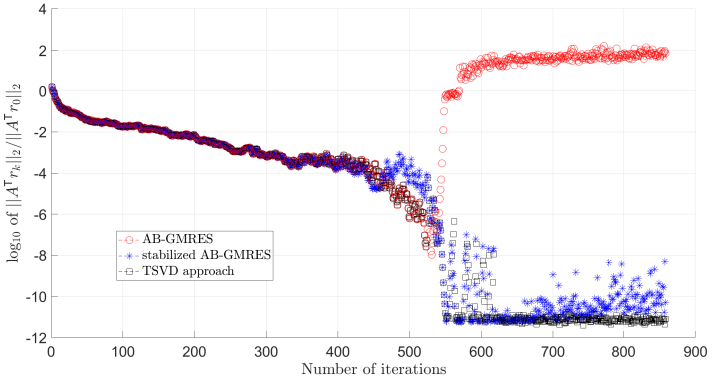


FIGURE 3.16: Comparison of the standard AB-GMRES with stabilized and TSVD stabilized AB-GMRES with $\mu = 10^{-8}$ for Maragal_3T.

TSVD method requires computing the truncated singular value decomposition of R_k, and requires choosing the value of the threshold parameter μ, whereas the stabilized AB-GMRES does not require either of them.

3.5.1.2 Comparison with Tikhonov regularization method

Another approach to stabilize AB-GMRES would be to apply Tikhonov regularization. There are two methods to implement it. The first method is to solve the following square system:

$$(R_k^{\mathsf{T}} R_k + \lambda I) y_k = R_k^{\mathsf{T}} t_k, \qquad \lambda \geq 0 \tag{3.55}$$

using the Cholesky decomposition. The second method is to solve the regularized least suqares problem

$$\min_{y_k \in \mathbb{R}^k} \left\| \begin{pmatrix} t_k \\ 0 \end{pmatrix} - \begin{pmatrix} R_k \\ \sqrt{\lambda} I \end{pmatrix} y_k \right\|_2 \tag{3.56}$$

using the QR decomposition. These two methods are equivalent mathematically. However, they are not equivalent numerically. The behavior of the first method is similar to the stabilized AB-GMRES.

Table 3.5 shows that AB-GMRES combined with the first method converges better when $\lambda = 10^{-16}$ than when $\lambda = 10^{-14}$ for the problem Maragal_3T. This method can be used to shift upwards the small singular values, but is less acurrate compared to the stabilized AB-GMRES.

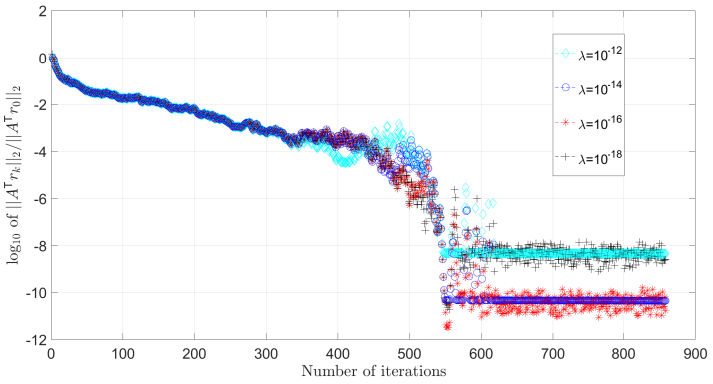


FIGURE 3.17: Relative residual norm for AB-GMRES with Tikhonov regularization using (3.56) versus number of iterations for different values of the regularization parameter λ for Maragal_3T.

Table 3.5 also shows that the second method is even more accurate compared with the stabilized AB-GMRES method. There is no need to form the normal equations, so that less information is lost due to rounding error. However, one needs to choose an appropriate value for the regularization parameter λ. Figure 3.17 shows the relative residual norm $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2$ for AB-GMRES with Tikhonov regularization using (3.56) versus the number of iterations for different values of λ for Maragal_3T. According to Figure 3.17, $\lambda = 10^{-16}$ was optimal among $10^{-12}, 10^{-14}, 10^{-16}$, and 10^{-18}.

We here note the following.

Theorem 3.17. *Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ be the singular values of R_k. Then, the singular values of*

$$R'_k = \begin{pmatrix} R_k \\ \sqrt{\lambda} I \end{pmatrix} \tag{3.57}$$

are given by $\sqrt{\sigma_1^2 + \lambda} \geq \sqrt{\sigma_2^2 + \lambda} \geq \cdots \geq \sqrt{\sigma_k^2 + \lambda}$.

Proof. Let the singular value decomposition of R_k be given by $R_k = U\Sigma V^{\mathsf{T}} \in \mathbb{R}^{k\times k}$, where U, V are orthogonal matrices and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$. Let $I_k \in \mathbb{R}^{k\times k}$ be the identity matrix. Then, we have $R'_k = \begin{pmatrix} R_k \\ \sqrt{\lambda} I_k \end{pmatrix} = U'\Sigma' V^{\mathsf{T}}$, where $U' = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}$ and $\Sigma' = \begin{pmatrix} \Sigma \\ \sqrt{\lambda} I_k \end{pmatrix}$. Since $\Sigma'^{\mathsf{T}}\Sigma' = \Sigma^2 + \lambda I_k = \mathrm{diag}(\sigma_1^2 + \lambda, \sigma_2^2 + \lambda, \ldots, \sigma_k^2 + \lambda)$, the singular values of $\begin{pmatrix} R_k \\ \sqrt{\lambda} I_k \end{pmatrix}$ are $\sqrt{\sigma_1^2 + \lambda} \geq \sqrt{\sigma_2^2 + \lambda} \geq \cdots \geq \sqrt{\sigma_k^2 + \lambda}$. □

TABLE 3.5: Attainable smallest relative residual norm $\|A^\mathsf{T} r_k\|_2/\|A^\mathsf{T} r_0\|_2$ for AB-GMRES with Tikhonov regularization using (3.55) and (3.56), and stabilized AB-GMRES for Maragal_3T.

matrix	Maragal_3T	Maragal_4T	Maragal_5T	Maragal_6T	Maragal_7T
iter.	552	597	1304	2440	1864
method (3.55) $\lambda = 10^{-14}$	5.08×10^{-11}	5.57×10^{-8}	1.05×10^{-5}	8.26×10^{-6}	4.53×10^{-6}
iter.	570	598	1226	2440	1864
method (3.55) $\lambda = 10^{-16}$	5.80×10^{-12}	5.59×10^{-8}	4.22×10^{-6}	8.26×10^{-6}	4.53×10^{-6}
iter.	553	547	1261	2937	2475
method (3.56) $\lambda = 1.6\times10^{-14}$	7.54×10^{-11}	5.59×10^{-8}	1.15×10^{-5}	9.12×10^{-6}	2.78×10^{-7}
iter.	551	547	1262	3037	2475
method (3.56) $\lambda = 10^{-16}$	3.37×10^{-12}	5.59×10^{-8}	5.64×10^{-7}	1.91×10^{-6}	2.78×10^{-7}
iter.	552	(&) 598	(∗) 1224	(∗) 3000	(∗) 2475
stabilized AB-GMRES	4.86×10^{-12}	5.59×10^{-8}	2.54×10^{-6}	4.56×10^{-6}	2.78×10^{-7}

Then, let

$$\kappa \equiv \kappa_2(R_k) = \frac{\sigma_1}{\sigma_k}, \quad \kappa'^2 \equiv \kappa_2(R'_k)^2 = \frac{\sigma_1^2+\lambda}{\sigma_1^2/\kappa^2+\lambda} = 1 + \frac{\sigma_1^2(1-1/\kappa^2)}{\sigma_1^2/\kappa^2+\lambda}. \tag{3.58}$$

Since $\kappa \geq 1, \mathrm{d}\kappa'/\mathrm{d}\lambda \leq 0$ for $\lambda \geq 0$ and $\kappa'(\lambda = 0) = \kappa, \kappa'(\lambda = +\infty) = 1$. Note also that

$$\lambda = \frac{\sigma_1^2[1+(\kappa'/\kappa)^2]}{\kappa'^2 - 1}. \tag{3.59}$$

Therefore, for instance, if $\kappa \gg 1$ and we want $\kappa' = \sqrt{\kappa}$,

$$\lambda = \frac{\sigma_1^2(1+1/\kappa)}{\kappa - 1} \simeq \frac{\sigma_1^2}{\kappa}. \tag{3.60}$$

For example, if $\kappa = 10^{16}$ and we want $\kappa' = 10^8$, we should choose $\lambda \simeq \sigma_1^2 \times 10^{-16}$. For Maragal_3T, the largest singular value σ_1 is about 12.64, so that we can estimate a reasonable value of $\lambda \simeq 1.60 \times 10^{-14}$. However, this estimation assumes $\kappa' = \sqrt{\kappa}$, and needs an extra cost for computing σ_1. See [34] for other estimation techniques for the regularization parameter.

3.5.1.3 Comparison with the Range Restricted GMRES

We compared the proposed stabilized AB-GMRES with the range restricted AB-GMRES (RR-AB-GMRES) [33], where the Krylov subspace for the RR-AB-GMRES with $B = A^\mathsf{T}$ is $\mathcal{K}_k(AA^\mathsf{T}, AA^\mathsf{T} r_0)$, and the standard AB-GMRES with $B = A^\mathsf{T}$.

TABLE 3.6: Comparison of the attainable smallest relative residual norm $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2$.

matrix	Maragal_3T	Maragal_4T	Maragal_5T	Maragal_6T	Maragal_7T
iter.	531	465	1110	2440	1864
standard AB-GMRES	1.05×10^{-8}	2.09×10^{-7}	5.35×10^{-6}	8.26×10^{-6}	4.53×10^{-6}
iter.	552	(&) 598	(∗) 1224	(∗) 3000	(∗) 2475
stabilized AB-GMRES	4.86×10^{-12}	5.59×10^{-8}	2.54×10^{-6}	4.56×10^{-6}	2.78×10^{-7}
iter.	553	565	1223	2374	2474
RR-AB-GMRES	2.57×10^{-11}	5.59×10^{-8}	3.62×10^{-6}	1.63×10^{-5}	2.78×10^{-7}

Table A.1 gives the number of iterations and the smallest relative residual norm for the RR-AB-GMRES, the standard and stabilized AB-GMRES for the Maragal matrices. The table shows that the stabilized AB-GMRES is more accurate than the standard AB-GMRES. Table A.1 also shows that the stabilized AB-GMRES is generally more accurate than the RR-AB-GMRES. The stabilized AB-GMRES took more iterations to attain the same order of the smallest residual norm than the RR-AB-GMRES.

3.5.2 Inconsistent systems with severely ill-conditioned range-symmetric coefficient matrices

Next, we test the stabilized AB-GMRES on least squares problems $\min_{x\in\mathbb{R}^n}\|b-Ax\|_2$ by GMRES, where $A\in\mathbb{R}^{n\times n}$ are severely ill-conditioned range-symmetric (square) matrices given in Table A.3.

These matrices are all numerically singular. We generated the right-hand side b by the MATLAB function `rand`, so that the systems are generically inconsistent. We compared the stabilized AB-GMRES with the standard AB-GMRES and RR-AB-GMRES. Table A.4 gives the smallest relative residual norm and the corresponding number of iterations. Table A.6 gives the CPU times in seconds required to obtain relative residual norm $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2 < 10^{-8}$. The switching strategy which was introduced in Section 3.4.1 was used for the stabilized AB-GMRES when measuing CPU times. The number of iterations when switching occurred is in brackets.

For Harvard500 and bw42, AB-GMRES could only converge to the level of 10^{-9} regarding the relative residual norm, while the stabilized AB-GMRES converged to the level of 10^{-14}. The stabilized AB-GMRES was robust in the sense that it could

TABLE 3.7: Information of the singular square matrices.

matrix	size	density[%]	rank	$\kappa_2(A)$	application
Harvard500	500	1.05	170	1.30×10^{2}	web connectivity
netz4504	1961	0.13	1342	3.41×10^{1}	2D/3D finite element problem
TS	2142	0.99	2140	3.52×10^{3}	counter example problem
grid2_dual	3136	0.12	3134	8.58×10^{3}	2D/3D finite element problem
uk	4828	0.06	4814	6.62×10^{3}	undirected graph
bw42	10000	0.05	9999	2.03×10^{3}	partial differential equation[19]
msc01050	1050	2.38	1049	1.31×10^{8}	2D/3D structural problem
freeFlyingRobot_7	3918	0.20	3881	1.68×10^{12}	optimal control problem

TABLE 3.8: Comparison of the attainable smallest relative residual norm $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2$ for inconsistent square linear systems.

matrix	Harvard500	netz4504	TS	grid2_dual	uk	bw42
iter. standard AB-GMRES	104 9.38×10^{-9}	144 4.51×10^{-10}	1487 1.56×10^{-9}	3134 5.98×10^{-10}	4620 1.35×10^{-9}	715 8.06×10^{-8}
iter. stabilized AB-GMRES	(∗) 134 8.46×10^{-14}	(&) 201 1.51×10^{-14}	1613 2.51×10^{-9}	(∗) 3135 5.53×10^{-10}	(∗) 4739 6.57×10^{-10}	(&) 788 1.66×10^{-7}
iter. RR-AB-GMRES	135 7.78×10^{-14}	200 3.36×10^{-14}	1652 4.56×10^{-9}	3134 6.52×10^{-8}	4706 8.33×10^{-8}	1163 1.56×10^{-5}

TABLE 3.9: Comparison of the CPU time (seconds) to obtain relative residual norm $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2 < 10^{-8}$ for inconsistent square linear systems.

matrix	Harvard500	netz4504	TS	grid2_dual	uk	bw42
iter.	104	134	1411	3134	4583	-
standard AB-GMRES	4.72×10^{-2}	1.87×10^{-1}	2.14×10	2.16×10^{2}	6.93×10^{2}	-
iter.	104	134	1531 (182)	3134	4679 (4199)	-
stabilized AB-GMRES	4.78×10^{-2}	1.89×10^{-1}	8.19×10	2.21×10^{2}	1.93×10^{3}	-
iter.	114	153	1530	-	-	-
RR-AB-GMRES	6.42×10^{-2}	2.62×10^{-1}	2.68×10	-	-	-

TABLE 3.10: Attainable smallest relative residual norm $\|A^{\mathsf{T}}r_k\|_2/\|A^{\mathsf{T}}r_0\|_2$ for range symmetric matrices.

matrix	bw42	msc01050	freeFlyingRobot_7
iter.	147	560	1084
standard GMRES	8.08×10^{-9}	4.98×10^{-8}	8.86×10^{-8}
iter.	219	668	3414
stabilized GMRES	2.11×10^{-11}	4.62×10^{-9}	3.24×10^{-10}
iter.	220	564	3183
RR-GMRES	3.13×10^{-11}	2.62×10^{-6}	1.40×10^{-9}

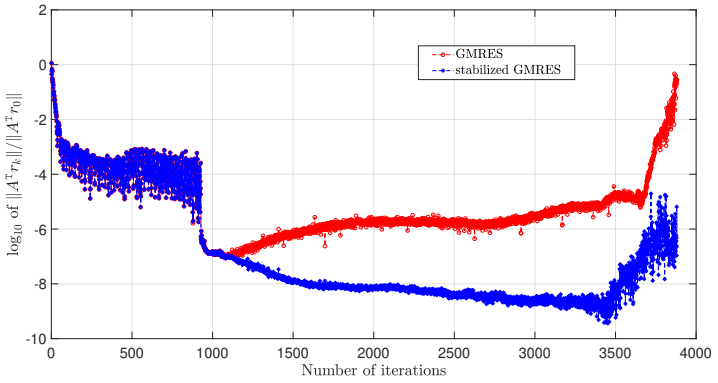


FIGURE 3.18: Comparison of GMRES with stabilized GMRES for freeFlyingRobot_7.

continue to compute even when the upper triangular matrix R_k became seriously ill-conditioned, and the relative residual norm did not increase sharply towards the end, but just stagnated at a low level, just like for consistent problems.

Thus, our stabilization method also makes AB-GMRES stable for highly ill-conditioned inconsistent systems with square coefficient matrices.

The coefficient matrix A of bw42 is singular and satisfies $\mathcal{N}(A) = \mathcal{N}(A^{\mathsf{T}})$. The problem comes from a finite-difference discritization of a PDE with periodic boundary condition (Experiment 4.2 in Brown and Walker[19] with the original b). Since the matrix is range symmetric, the GMRES, RR-GMRES, and stabilized GMRES can be directly applied to $Ax = b$ (See [19] Theorem 2.4, [35] Theorem 2.7, and [20] Theorem 3.2.) as shown in Table A.5. The stabilized GMRES gave a relative residual norm 1.94×10^{-11} for bw42 at the 219th iteration. The proposed method can be considered as a way of making the GMRES stable for highly ill-conditioned inconsistent problems.

Figure 3.18 shows comparison of GMRES with stabilized GMRES for a symmetric

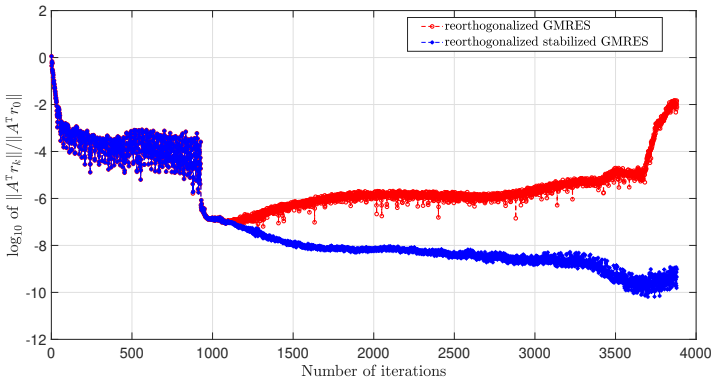


FIGURE 3.19: Comparison of reorthogonalized GMRES with reorthogonalized stabilized GMRES for freeFlyingRobot_7.

matrix (, which is range symmetric), freeFlyingRobot_7 which contains a cluster of tiny singular values which gradually decrease to zero. The stabilized GMRES converged to 3.65×10^{-10} at 3,452 iterations, better than GMRES. But the relative residual increased after the 3,452 iterations. Hence, we adopted a reorthogonalization strategy which performs the modified Gram-Schmidt orthogonalization process once more. We replaced line 4-6 of Algorithm 4 (GMRES version) by Algorithm 6 to reorthogonalize GMRES and the stabilized GMRES. As in Figure 3.19, after reorthogonalization, the stabilized GMRES became more stabilized and converged to a relative residual of 6.45×10^{-11} at 3,701 iterations.

Algorithm 6 reorthogonalized modified Gram-Schmidt

1: **for** $i = 1, 2$ **do**
2: **for** $j = 1, 2, \ldots, k$ **do**
3: $h_{j,k} = w_k^{\mathsf{T}} v_j, \quad w_k = w_k - h_{j,k} v_j$
4: **end for**
5: **end for**

3.6 Concluding Remarks

We proposed a stabilized AB-GMRES method for ill-conditioned underdetermined and inconsistent least squares problems. It shifts upwards the tiny singular values of the upper triangular matrix appearing in AB-GMRES, making the process more stable, giving better convergence, and more accurate solutions compared to AB-GMRES. We have also given a theoretical analysis to explain why the proposed method works. The

method is also effective for making GMRES stable for range-symmetric inconsistent least squares problems with severely ill-conditioned square coefficient matrices.

Chapter 4

Convergence analysis of inner-iteration preconditioned GMRES

This chapter first introduces BAG-GMRES and NR-SOR. Then, it reviews inner-iteration preconditioned GMRES. Then, it analyzes the numerical example and analyzes the convergence of inner-iteration GMRES. [36, 37]

4.1 Previous work

4.1.1 BA-GMRES method

Consider solving the overdetermined least squares problem

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_2, \qquad A \in \mathbb{R}^{m\times n}, \qquad b \in \mathbb{R}^m, \qquad m > n, \tag{4.1}$$

by BA-GMRES [8], where $B \in \mathbb{R}^{n\times m}$ is a preconditioning matrix, which is equivalent to applying GMRES to

$$BAx = Bb, \qquad A \in \mathbb{R}^{m\times n}, \qquad B \in \mathbb{R}^{n\times m}, \qquad b \in \mathbb{R}^m, \tag{4.2}$$

if $\mathcal{R}(B^\mathsf{T}BA) = \mathcal{R}(A)$. The algorithm is as follows.

Algorithm 7 BA-GMRES

Choose $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0$, , $w_0 = Br_0$, $v_1 = w_0/\|w_0\|_2$,
for $i = 1, 2, \ldots, k$ **do**
    $w_i = BAv_i$,
    for $j = 1, 2, \ldots, i$ **do**
        $h_{i,j} = w_i^{\mathsf{T}} v_j$, $w_i = w_i - h_{j,i} v_j$,
    end for
    $h_{i+1,i} = \|w_i\|_2$, $v_{i+1} = w_i/h_{i+1,i}$,
    Compute $y_i \in \mathbb{R}^i$ which minimizes $\|w_i\|_2 = \|\|w_0\|_2 e_1 - H_{i+1,i} y_i\|_2$,
    $x_i = x_0 + [v_1, v_2, \ldots, v_i] y_i$, $r_i = b - Ax_i$.
    if $\|A^{\mathsf{T}} r_i\|_2 < \epsilon \|A^{\mathsf{T}} r_0\|_2$ **then**
        stop
    end if
end for

4.1.2 Stationary iterative method

Stationary iterative methods are a type of iterative solvers for systems of linear equations. Stationary means a fixed iterative scheme. Thus, it is defined as $x_{k+1} = \Phi(x_k)$, which needs an initial guess x_0 to start the iteration. Denote the exact solution of the system of linear equations $Ax = b$, with a square coefficient matrix $A \in \mathbb{R}^{n \times n}$ as x^*. Denote the error of the kth iterate as $e_k = x_k - x^*$. A stationary iterative method is called linear if there exists a matrix $C \in \mathbb{R}^{n \times n}$ that satisfies

$$e_{k+1} = Ce_k. \tag{4.3}$$

Denote the spectral radius of C as $\rho(C)$, i.e. $\rho(C) = \max\{|\lambda_i| \big| \ \lambda_i : \text{eigenvalue of } C\}$. If $\rho(C) < 1$, then $\lim_{k \to +\infty} C^k = 0$, which can ensure $\lim_{k \to +\infty} e_{k+1} = \lim_{k \to +\infty} C^k e_1 = 0$. Thus, the linear stationary iterative method converges if and only if $\rho(C) < 1$.

One basic idea of the linear stationary iterative method is matrix splitting of A.

$$A = M - N, \tag{4.4}$$

where M is easily invertible. Then, $Ax = b$ is $(M - N)x = Mx - Nx = b$. Moving Nx to the right-hand side, we obtain $Mx = Nx + b$. Then, we construct an iterative scheme

$$Mx_{k+1} = Nx_k + b. \tag{4.5}$$

If, M is invertible, we have

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b, \tag{4.6}$$

and

$$x^* = M^{-1}Nx^* + M^{-1}b. \tag{4.7}$$

Then, we obtain

$$\begin{aligned} e_{k+1} &= x_{k+1} - x^* && (4.8)\\ &= (M^{-1}Nx_k + M^{-1}b) - (M^{-1}Nx^* + M^{-1}b) && (4.9)\\ &= M^{-1}N(x_k - x^*) && (4.10)\\ &= M^{-1}Ne_k. && (4.11) \end{aligned}$$

Accoriding to (4.3), $C = M^{-1}N$.

Richardson's method, Jacobi method, Gauss-Seidel method, successive over-relaxation method (SOR), symmetric successive over-relaxation (SSOR) and Hermitian / skew-Hermitian splitting (HSS) method are linear stationary iterative methods based on different choices of M and N. By choosing proper parameters for each method, the condition $\rho(M^{-1}N) < 1$ is ensured.

If $x_0 = 0$, from (4.6), we have

$$x_1 = M^{-1}b. \tag{4.12}$$

$$x_2 = (M^{-1}N + \mathrm{I})M^{-1}b. \tag{4.13}$$

$$x_3 = ((M^{-1}N)^2 + M^{-1}N + \mathrm{I})M^{-1}b. \tag{4.14}$$

$$x_l = (\sum_{i=1}^{l-1}(M^{-1}N)^i + \mathrm{I})M^{-1}b, \qquad l \geq 1. \tag{4.15}$$

4.1.3 NR-SOR method

Splitting the matrix A into three parts gives

$$A = D + L + U, \tag{4.16}$$

where D is the diagonal part, L is the strictly lower triangular part, and, U is the strictly upper triangular part of A, respectively. The successive over-relaxation (SOR) method [3] chooses

$$M = \frac{1}{\omega}D + L, \qquad N = (\frac{1}{\omega} - 1)D - U. \tag{4.17}$$

NR-SOR[38–40] is equivalent to applying SOR to the normal equation

$$A^{\mathsf{T}}Ax = A^{\mathsf{T}}b. \tag{4.18}$$

as

$$(M - N)x = A^{\mathsf{T}}b \tag{4.19}$$

The algorithm is given below.

Let a_i be the ith column of A, $i = 1, 2, \ldots, n$. Suppose $a_i \neq 0, i = 1, 2, \ldots, n$.

Algorithm 8 NR-SOR

Let x^0 be the initial solution and $r = b - Ax^0$, $0 < \omega < 2$.
for $k = 1, 2, \ldots, l$ **do**
    for $i = 1, 2, \ldots, n$ **do**
        $\delta_i = \omega(r, a_i)/||a_i||_2^2$,
        $x_i^{k+1=}x_i^k + \delta_i$,
        $r = r - \delta_i a_i$.
    end for
end for

Note that, the computation of $||a_i||_2^2$ is done only once in the beginning.

4.1.4 Inner-iteration GMRES

Set $B = A^{\mathsf{T}}$ in BA-GMRES, which is equivalent to applying GMRES to the normal equations

$$A^{\mathsf{T}}Ax = A^{\mathsf{T}}b. \tag{4.20}$$

. One can precondition this system by an explicit matrix $P \in \mathbb{R}^{n\times n}$, which is

$$PA^{\mathsf{T}}Ax = PA^{\mathsf{T}}b. \tag{4.21}$$

. Forming an explicit matrix P needs time and storage space, especially when you need to form the normal equation matrix $A^{\mathsf{T}}A$ explicitly.

Applying NR-SOR to the normal equations for l steps, which avoids forming the normal equation matrix $A^{\mathsf{T}}A$ of (4.20) explicitly, is equivalent to providing a preconditioning matrix $P^{(l)}$ such that

$$P^{(l)}A^{\mathsf{T}}Ax = P^{(l)}A^{\mathsf{T}}b. \tag{4.22}$$

Introducing a stationary iteration method inside the GMRES iteration instead of forming an explicit preconditioning matrix to precondition GMRES, gives the inner-iteration preconditioned GMRES. Morikuni [9] did lots of work on different stationary iteration methods combined with AB-GMRES or BA-GMRES and compared with other methods.

As other earlier work, we mention FGMRES [41] which is more related to AB-GMRES but applying different preconditioners at each step. SOR was used as inner preconditioners with GCR [42], and SOR as inner preconditioners with GMRES[43, 44].

Using NR-SOR as inner-iteration preconditions, is a way of implicit preconditioning, but has an explicit form for theoretical analysis. In NR-SOR $A^{\mathsf{T}}A = M - N$, from (4.15)

$$\begin{aligned}
P^{(l)}A^{\mathsf{T}}A &= (\sum_{i=1}^{l-1}(M^{-1}N)^i + \mathrm{I})M^{-1}A^{\mathsf{T}}A && (4.23)\\
&= (\sum_{i=1}^{l-1}(M^{-1}N)^i + \mathrm{I})M^{-1}(M-N) && (4.24)\\
&= (\sum_{i=1}^{l-1}(M^{-1}N)^i + \mathrm{I})(\mathrm{I} - M^{-1}N) && (4.25)\\
&= \sum_{i=1}^{l-1}(M^{-1}N)^i + \mathrm{I} - \sum_{i=1}^{l}(M^{-1}N)^i && (4.26)\\
&= \mathrm{I} - (M^{-1}N)^l && (4.27)\\
&= \mathrm{I} - (H)^l && (4.28)
\end{aligned}$$

where $H = M^{-1}N$.

The algorithm for using NR-SOR as inner-iteration preconditioner in BA-GMRES is as follows.

Algorithm 9 NR-SOR inner-iteration BA-GMRES

Choose $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0$,
apply l steps SOR to $A^\mathsf{T}Aw = A^\mathsf{T}r_0$ to obtain $w_0 = P^l A^\mathsf{T} r_0$, (NR-SOR),
$v_1 = w_0/\|w_0\|_2$,
for $i = 1, 2, \ldots, k$ **do**
    $u_i = Av_i$,
    apply l steps SOR to $A^\mathsf{T}Aw = A^\mathsf{T}u_i$ to obtain $w_i = P^l A^\mathsf{T} u_i$, (NR-SOR),
    for $j = 1, 2, \ldots, i$ **do**
        $h_{i,j} = w_i^\mathsf{T} v_j$, $w_i = w_i - h_{j,i}v_j$,
    end for
    $h_{i+1,i} = \|w_i\|_2$, $v_{i+1} = w_i/h_{i+1,i}$,
    Compute $y_i \in \mathbb{R}^i$ which minimizes $\|w_i\|_2 = \|\|w_0\|_2 e_1 - H_{i+1,i}y_i\|_2$,
    $x_i = x_0 + [v_1, v_2, \ldots, v_i]y_i$, $r_i = b - Ax_i$.
    if $\|A^\mathsf{T}r_i\|_2 < \epsilon\|A^\mathsf{T}r_0\|_2$ **then**
        stop
    end if
end for

4.1.5 Convergence of NR-SOR method

Theorem 4.1. [45] *If A is symmetric positive definite and $0 < \omega < 2$, then, $\rho(M^{-1}N) < 1$ for the SOR method.*

Proof. A is symmetric, $U = L^\mathsf{T}$.

$$A = L + D + L^\mathsf{T}, \qquad M = \frac{1}{\omega}D + L, \qquad N = -[(1 - \frac{1}{\omega})D + L^\mathsf{T}]. \tag{4.29}$$

Denote the eigenvalue of $-M^{-1}N$ as λ, and v as corresponding eigenvector.

$$(\frac{1}{\omega}D + L)^{-1}[(1 - \frac{1}{\omega})D + L^\mathsf{T}]v = \lambda v. \tag{4.30}$$

$$[(1 - \frac{1}{\omega})D + L^\mathsf{T}]v = \lambda(\frac{1}{\omega}D + L)v. \tag{4.31}$$

$$[(\omega - 1)D + \omega L^\mathsf{T}]v = \lambda(D + \omega L)v. \tag{4.32}$$

Let v^H denote the conjugate transpose of v.

$$v^\mathsf{H}[(\omega - 1)D + \omega L^\mathsf{T}]v = \lambda v^\mathsf{H}(D + \omega L)v. \tag{4.33}$$

A is symmetric positive definite. Thus,

$$p = v^\mathsf{H} D v > 0. \tag{4.34}$$

Let

$$v^{\mathsf{H}} L v = \alpha + i\beta, \tag{4.35}$$

then,

$$v^{\mathsf{H}} L^{\mathsf{T}} v = v^{\mathsf{H}} L^{\mathsf{H}} v = (v^{\mathsf{H}} L v)^{\mathsf{H}} = \alpha - i\beta. \tag{4.36}$$

Based on (4.34), (4.35), and (4.36),

$$v^{\mathsf{H}}[(\omega-1)D + \omega L^{\mathsf{T}}]v = (\omega-1)v^{\mathsf{H}} D v + \omega v^{\mathsf{H}} L^{\mathsf{T}} v = (\omega-1)p + \omega(\alpha - i\beta). \tag{4.37}$$

$$v^{\mathsf{H}}(D + \omega L)v = v^{\mathsf{H}} D v + \omega v^{\mathsf{H}} L v = p + \omega(\alpha + i\beta). \tag{4.38}$$

A is symmetric positive definite, thus,

$$v^{\mathsf{H}} A v = v^{\mathsf{H}}(L^{\mathsf{T}} + L + D)v = p + 2\alpha > 0. \tag{4.39}$$

Due to $p > 0$, $0 < \omega < 2$, and $p + 2\alpha > 0$,

$$p + \omega\alpha = (1 - \frac{\omega}{2})p + \frac{\omega}{2}(p + 2\alpha) > 0. \tag{4.40}$$

Thus,

$$v^{\mathsf{H}}(D + \omega L)v = (p + \omega\alpha) + i\omega\beta \neq 0. \tag{4.41}$$

From (4.33) and (4.41),

$$\lambda = \frac{v^{\mathsf{H}}[(\omega-1)D + \omega L^{\mathsf{T}}]v}{v^{\mathsf{H}}(D + \omega L)v}. \tag{4.42}$$

From (4.37) and (4.38)

$$\lambda = \frac{v^{\mathsf{H}}[(\omega-1)D + \omega L^{\mathsf{T}}]v}{v^{\mathsf{H}}(D + \omega L)v} = \frac{[(\omega-1)p + \omega\alpha] - i\omega\beta}{(p + \omega\alpha) + i\omega\beta} \tag{4.43}$$

$$|\lambda|^2 = \lambda^{\mathsf{H}}\lambda = \frac{[(\omega-1)p + \omega\alpha]^2 + \omega^2\beta^2}{(p + \omega\alpha)^2 + \omega^2\beta^2} \tag{4.44}$$

$$(p + \omega\alpha)^2 - [(\omega-1)p + \omega\alpha]^2 = \{p + \omega\alpha + [(\omega-1)p + \omega\alpha]\}\{p + \omega\alpha - [(\omega-1)p + \omega\alpha]\} \tag{4.45}$$

$$= \omega p(p + 2\alpha)(2 - \omega) \tag{4.46}$$

$$> 0. \tag{4.47}$$

Thus,

$$|\lambda|^2 = \frac{[(\omega-1)p+\omega\alpha]^2+\omega^2\beta^2}{(p+\omega\alpha)^2+\omega^2\beta^2} < 1. \tag{4.48}$$

As a conclusion, $|\lambda| < 1$, which means $\rho(M^{-1}N) < 1$. Hence, the SOR method converges. □

If $A^{\mathsf{T}}A$ is symmetric positive definite and $0 < \omega < 2$, NR-SOR converges and $\rho(M^{-1}N) < 1$ from Theorem 4.1. If $A^{\mathsf{T}}A$ is symmetric semi-definite, we require the semi-convergence of $M^{-1}N$ [10]. Thus, $\rho(M^{-1}N) \leq 1$.

4.1.6 Convergence of NR-SOR inner-iteration preconditioned BA-GMRES

NR-SOR inner-iteration BA-GMRES has an effect of speeding up the convergence of BA-GMRES. [9]

To understand this effect, let us look at the following, where $B^{(l)} = P^{(l)}A^{\mathsf{T}}$.

$$\|B^{(l)}r_k\|_2 = \min_{p_k} \|p_k(B^{(l)}A)B^{(l)}r_0\|_2 \tag{4.49}$$

Here, p_k is a polynomial of degree $\leq k$ which satisfies $p_k(0) = 1$. An upper bound for $\|B^{(l)}r_k\|_2$ was obtained using the spectral radius of $H = M^{-1}N$ [10], where $B^{(l)}A = \mathrm{I} - H^l$. However, the bound is pessimistic and does not explain the observed fast convergence.

Since $\rho(H) \leq 1$, the eigenvalues of $B^{(l)}A$ approach 1 as l increases. The eigenvalues of $B^{(l)}A$ cluster but the spectral radius of $\rho(H^l)$ only changes a little. After l steps of inner-iteration preconditioning, their exists a cluster of eigenvalues near 1. Later we will focus on the distribution of eigenvalues to analyze the convergence.

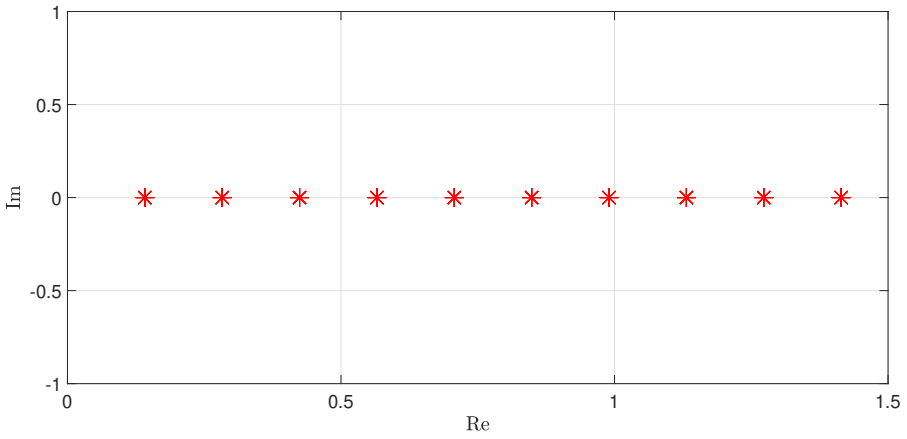


FIGURE 4.1: The nonzero singular values of the test matrix A.

4.1.7 Numerical example

We use a test matrix [9] to introduce our analysis.

$$A = U \begin{pmatrix} 1 & 1 & & & & & 0 \\ & & 0.9 & 0.9 & & & \\ & & & & \ddots & \ddots & \\ & & & & & 0.1 & 0.1 \\ 0 & & & & & & \end{pmatrix} V^{\mathsf{T}} \in R^{100\times 20}. \tag{4.50}$$

where $U \in \mathbb{R}^{100\times 100}$ and $V \in \mathbb{R}^{20\times 20}$ are orthogonal matrices computed with the QR factorization of random matrices. Thus, A is rank-deficient, with rank 10.

Figure 4.1 shows the nonzero singular values of A. Figure 4.2 shows the nonzero eigenvalue of $A^{\mathsf{T}}A$. Figure 4.3 shows the nonzero eigenvalues of $H = M^{-1}N$. Figre 4.4 show the eigenvalues of $B^{(l)}A = P^{(l)}A^{\mathsf{T}}A = \mathrm{I} - H^l, l = 4$. Figre 4.5 show the eigenvalues of $B^{(l)}A = P^{(l)}A^{\mathsf{T}}A = \mathrm{I} - H^l, l = 8$. Table 4.1 includes the values of above figures.

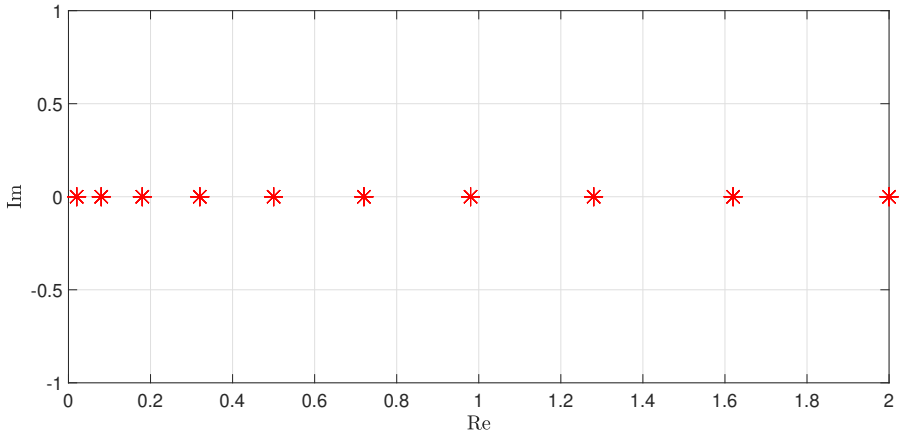


FIGURE 4.2: The nonzero eigenvalues of the normal equation matrix $A^{\mathsf{T}}A$ of the test matrix A.

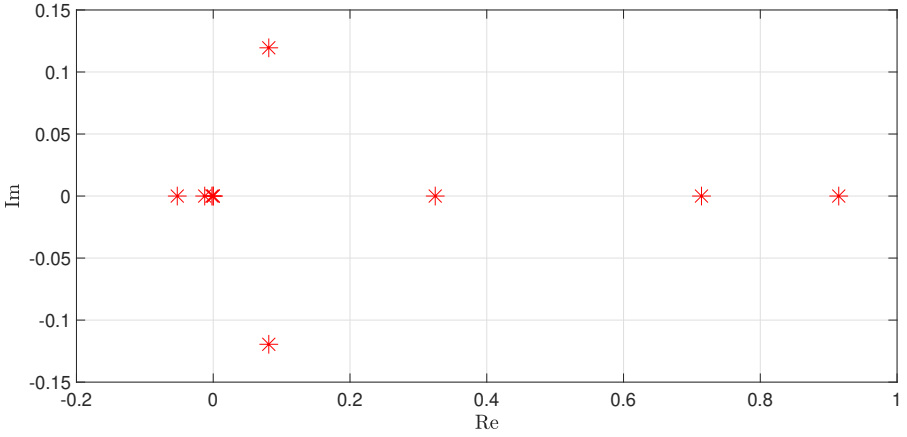


FIGURE 4.3: The nonzero eigenvalues of $H = M^{-1}N$ of the test matrix A.

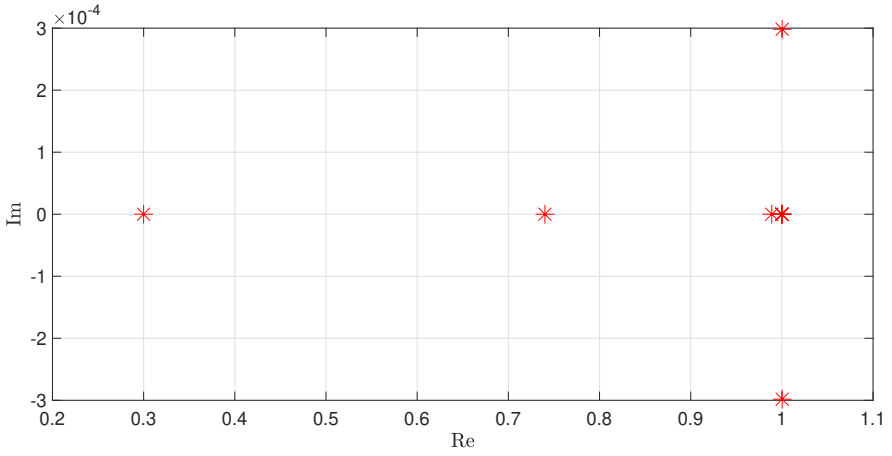


FIGURE 4.4: The nonzero eigenvalues of

$$B^{(l)}A = \mathrm{I} - H^{l}(l = 4)$$

of the test matrix A.

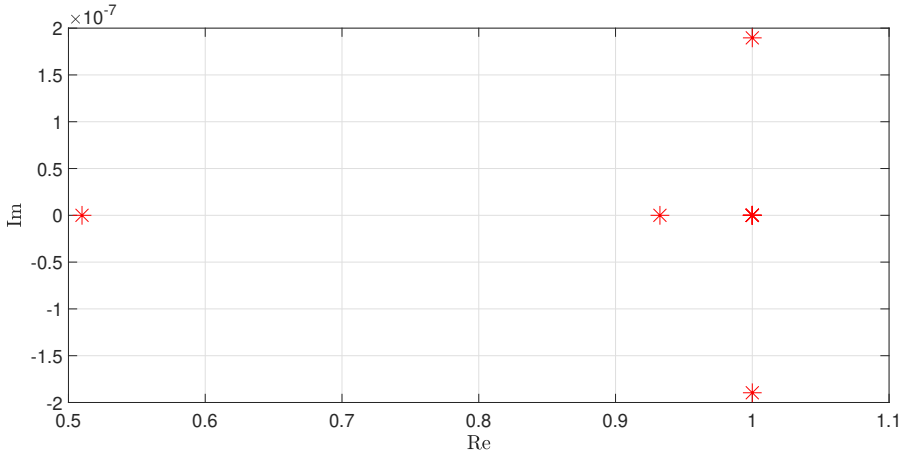


FIGURE 4.5: The nonzero eigenvalues of $B^{(l)}A = \mathrm{I} - H^l(l = 8)$ of the test matrix A.

TABLE 4.1: The singular values of A, eigenvalues of $A^{\mathsf{T}}A$, $H(M^{-1}N)$, and $B^{(l)}A = \mathrm{I} - H^l(l = 4, 8)$.

	A	$A^{\mathsf{T}}A$	$H = M^{-1}N$	$B^{(l)}A = \mathrm{I} - H^l(l = 4)$	$B^{(l)}A = \mathrm{I} - H^l(l = 8)$
1	1.41	2.00	0.00	1.00	1.00
2	1.27	1.62	0.00	1.00	1.00
3	1.31	1.28	0.00	1.00	1.00
4	0.99	0.98	0.01	1.00	1.00
5	0.85	0.72	0.05	1.00	1.00
6	0.71	0.50	$0.08 + 0.12i$	$1.00 + 2.98 \times 10^{-4}i$	$1.00 + 1.90 \times 10^{-7}i$
7	0.57	0.32	$0.08 - 0.12i$	$1.00 - 2.98 \times 10^{-4}i$	$1.00 - 1.90 \times 10^{-7}i$
8	0.42	0.18	0.32	0.99	1.00
9	0.28	0.08	0.71	0.74	0.93
10	0.14	0.02	0.91	0.30	0.51

4.2 Convergence analysis

Before starting our analysis, we mention that Ipsen used the Vandermonde matrix to analyze the convergence for GMRES in [46], and for multiple clusters case in [47]. The main difference is that we start from digonalizable matrices instead of normal matrices and used different techniques. Our analysis is an estimation of the convergence curve of the GMRES, and not an upper bound. Moreover, our analysis gives a clearer way to show how the number of clusters and their radii affect the convergence.

4.2.1 Convergence analysis of the test problem

Consider the problem

$$\widetilde{A}x = \widetilde{b}, \tag{4.51}$$

where

$$\widetilde{A} = P^{(l)}A^{\mathsf{T}}A = B^{(l)}A = \mathrm{I} - H^l \tag{4.52}$$

is an $n \times n$ square matrix, and

$$\widetilde{b} = P^{(l)}A^{\mathsf{T}}b = B^{(l)}b. \tag{4.53}$$

Let $x_0 = 0$ be the initial solution, then, $\widetilde{b}$ is the initial residual.
we use a Krylov subspace,

$$K_k(\widetilde{A}, \widetilde{b}) \equiv \mathrm{span}\{\widetilde{b}, \widetilde{A}\widetilde{b}, \cdots, \widetilde{A}^{k-1}\widetilde{b}\}, \tag{4.54}$$

to obtain an approximate solution $x_k \in K_k(\widetilde{A}, \widetilde{b})$ of (4.51).

If $\widetilde{b}$ only contains v_1, then Av_1, shares the same direction with v_1, which means $K_1(\widetilde{A}, \widetilde{b})$ reaches A-invariance, and the grade of $K_k(\widetilde{A}, \widetilde{b})$ is one. If $\widetilde{b}$ contains v_1 and v_2, then $K_2(\widetilde{A}, \widetilde{b})$ reaches A-invariance, and the grade of $K_k(\widetilde{A}, \widetilde{b})$ is two. Other cases have similar results.

Assume that $\widetilde{A}$ is diagonalizable, and d is the grade of $\mathcal{K}_k(\widetilde{A},\widetilde{b})$, which means d is the smallest integer such that $K_d(\widetilde{A}, \widetilde{b}) = K_{d+1}(\widetilde{A}, \widetilde{b})$.

Thus, $\widetilde{b}$ is spanned by the eigenvectors $v_1, v_2, \ldots, v_d$ corresponding to distinct eigenvalues of $\lambda_1, \lambda_2, \ldots, \lambda_d$.

Let, $V_d = [v_1, \cdots, v_d]$, and

$$\widetilde{b} = V_d[1, \cdots, 1]^{\mathsf{T}}. \tag{4.55}$$

$$\widetilde{A}\widetilde{b} = V_d[\lambda_1, \cdots, \lambda_d]^{\mathsf{T}} \tag{4.56}$$

$$\widetilde{A}^k\widetilde{b} = V_d[\lambda_1^k, \cdots, \lambda_d^k]^{\mathsf{T}} \tag{4.57}$$

TABLE 4.2: The eigenvalues distribution of $\widetilde{A} = B^{(l)}A = \mathrm{I} - H^l (l = 8)$

eigenvalues	structure	value
λ_1	$1 + \epsilon_1$	$1 + 8.00 \times 10^{-15}$
λ_2	$1 + \epsilon_2$	$1 + 3.11 \times 10^{-15}$
λ_3	$1 + \epsilon_3$	$1 + 2.44 \times 10^{-15}$
λ_4	$1 + \epsilon_4$	$1 + 5.86 \times 10^{-11}$
λ_5	1	1
λ_6	λ_6	$1.00 + 1.90 \times 10^{-7}i$
λ_7	λ_7	$1.00 - 1.90 \times 10^{-7}i$
λ_8	λ_8	0.9999
λ_9	λ_9	0.9325
λ_{10}	λ_{10}	0.5099

Let $x_k = (\widetilde{b}, \widetilde{A}\widetilde{b}, \cdots, \widetilde{A}^{k-1}\widetilde{b})(y_1, y_2, \ldots, y_k)^{\mathsf{T}} \in K_k(\widetilde{A}, \widetilde{b})$. Then,

$$\widetilde{b} - \widetilde{A}x_k = V_d[1, \cdots, 1]^{\mathsf{T}} - \widetilde{A}[\widetilde{b}, \widetilde{A}\widetilde{b}, \cdots, \widetilde{A}^{k-1}\widetilde{b}]y = V_d[1, \cdots, 1]^{\mathsf{T}} - V_d\Lambda_d^k y \tag{4.58}$$

$$\|\widetilde{b} - \widetilde{A}x_k\|_2 = \|V_d(\Lambda_d^k y - [1, \cdots, 1]^{\mathsf{T}})\|_2 \tag{4.59}$$

where

$$\Lambda_d^k = \begin{pmatrix} \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^k \\ \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^k \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_d & \lambda_d^2 & \cdots & \lambda_d^k \end{pmatrix} \in \mathbb{R}^{d \times k}. \tag{4.60}$$

As for the test matrix of (4.50) $A \in \mathbb{R}^{100 \times 20}$, $\widetilde{A} = B^{(l)}A = P^l A^{\mathsf{T}} A = \mathrm{I} - H^l$ has only one cluster of eigenvalues around the center 1, and the others are separate eigenvalues as shown in Figure 4.4 and 4.5 for $l = 4, 8$. Thus, according to Table 4.2 where $d = 10$

(4.60) is

$$\Lambda_{10} = \begin{pmatrix} 1+\epsilon_1 & (1+\epsilon_1)^2 & \cdots & (1+\epsilon_1)^{10} \\ 1+\epsilon_2 & (1+\epsilon_2)^2 & \cdots & (1+\epsilon_2)^{10} \\ 1+\epsilon_3 & (1+\epsilon_3)^2 & \cdots & (1+\epsilon_3)^{10} \\ 1+\epsilon_4 & (1+\epsilon_4)^2 & \cdots & (1+\epsilon_4)^{10} \\ 1 & 1 & \cdots & 1 \\ \lambda_6 & \lambda_6^2 & \cdots & \lambda_6^{10} \\ \lambda_7 & \lambda_7^2 & \cdots & \lambda_7^{10} \\ \lambda_8 & \lambda_8^2 & \cdots & \lambda_8^{10} \\ \lambda_9 & \lambda_9^2 & \cdots & \lambda_9^{10} \\ \lambda_{10} & \lambda_{10}^2 & \cdots & \lambda_{10}^{10} \end{pmatrix} \in \mathbb{R}^{10\times 10}. \tag{4.61}$$

At step $k < d$,

$$\Lambda_\epsilon = \begin{pmatrix} 1+\epsilon_1 & (1+\epsilon_1)^2 & \cdots & (1+\epsilon_1)^{k} \\ 1+\epsilon_2 & (1+\epsilon_2)^2 & \cdots & (1+\epsilon_2)^{k} \\ 1+\epsilon_3 & (1+\epsilon_3)^2 & \cdots & (1+\epsilon_3)^{k} \\ 1+\epsilon_4 & (1+\epsilon_4)^2 & \cdots & (1+\epsilon_4)^{k} \\ 1 & 1 & \cdots & 1 \\ \lambda_6 & \lambda_6^2 & \cdots & \lambda_6^{k} \\ \lambda_7 & \lambda_7^2 & \cdots & \lambda_7^{k} \\ \lambda_8 & \lambda_8^2 & \cdots & \lambda_8^{k} \\ \lambda_9 & \lambda_9^2 & \cdots & \lambda_9^{k} \\ \lambda_{10} & \lambda_{10}^2 & \cdots & \lambda_{10}^{k} \end{pmatrix} \in \mathbb{R}^{10\times k}. \tag{4.62}$$

Since, $\epsilon = \max_k |\epsilon_k| < 10^{-10}, k = 1, 2, 3, 4,$ which is very tiny,

$$\Lambda_\epsilon \approx \widetilde{\Lambda_\epsilon} = \begin{pmatrix} 1+\epsilon_1 & 1+2\epsilon_1 & \cdots & 1+k\epsilon_1 \\ 1+\epsilon_2 & 1+2\epsilon_2 & \cdots & 1+k\epsilon_2 \\ 1+\epsilon_3 & 1+2\epsilon_3 & \cdots & 1+k\epsilon_3 \\ 1+\epsilon_4 & 1+2\epsilon_4 & \cdots & 1+k\epsilon_4 \\ 1 & 1 & \cdots & 1 \\ \lambda_6 & \lambda_6^2 & \cdots & \lambda_6^k \\ \lambda_7 & \lambda_7^2 & \cdots & \lambda_7^k \\ \lambda_8 & \lambda_8^2 & \cdots & \lambda_8^k \\ \lambda_9 & \lambda_9^2 & \cdots & \lambda_9^k \\ \lambda_{10} & \lambda_{10}^2 & \cdots & \lambda_{10}^k \end{pmatrix} \in \mathbb{R}^{10\times k}. \tag{4.63}$$

Seperating $\widetilde{\Lambda_\epsilon}$ into two matrices, $\widetilde{\Lambda_\epsilon} = \Lambda_s + P$, where

$$\Lambda_s = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \lambda_6 & \lambda_6^2 & \cdots & \lambda_6^k \\ \lambda_7 & \lambda_7^2 & \cdots & \lambda_7^k \\ \lambda_8 & \lambda_8^2 & \cdots & \lambda_8^k \\ \lambda_9 & \lambda_9^2 & \cdots & \lambda_9^k \\ \lambda_{10} & \lambda_{10}^2 & \cdots & \lambda_{10}^k \end{pmatrix} \in \mathbb{R}^{10\times k}, \qquad P = \begin{pmatrix} \epsilon_1 & 2\epsilon_1 & \cdots & k\epsilon_1 \\ \epsilon_2 & 2\epsilon_2 & \cdots & k\epsilon_2 \\ \epsilon_3 & 2\epsilon_3 & \cdots & k\epsilon_3 \\ \epsilon_4 & 2\epsilon_4 & \cdots & k\epsilon_4 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{10\times k}. \tag{4.64}$$

$$\widetilde{\Lambda_s} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_6 & \lambda_6^2 & \cdots & \lambda_6^k \\ \lambda_7 & \lambda_7^2 & \cdots & \lambda_7^k \\ \lambda_8 & \lambda_8^2 & \cdots & \lambda_8^k \\ \lambda_9 & \lambda_9^2 & \cdots & \lambda_9^k \\ \lambda_{10} & \lambda_{10}^2 & \cdots & \lambda_6^k \end{pmatrix} \in \mathbb{R}^{6\times k}. \tag{4.65}$$

$$k = 6 \Rightarrow \det\widetilde{\Lambda_s} = \prod_{6\leq \mathrm{i}<\mathrm{j}\leq 10} (\lambda_\mathrm{i} - \lambda_\mathrm{j}) \prod_{\mathrm{i}=6}^{10} (\lambda_\mathrm{i} - 1). \tag{4.66}$$

Because $\lambda_i \neq \lambda_j \neq 1 \neq 0$, $(6 \leq i < j \leq 10)$,

$$\det\widetilde{\Lambda_s} \neq 0 \Rightarrow \mathrm{rank}\Lambda_\mathrm{s} = 6 \quad (\mathrm{k} = 6). \tag{4.67}$$

Note $\|V(\Lambda_s y - [1, 1, \cdots, 1]^\mathsf{T})\|_2 \leq \|V\|_2 \|\Lambda_s y - [1, 1, \cdots, 1]^\mathsf{T}\|_2$ in general.

$\|\widetilde{\Lambda_s} y - [1, 1, \cdots, 1]^\mathsf{T}\|_2 = 0$ and $\|\Lambda_s y - [1, 1, \cdots, 1]^\mathsf{T}\|_2 = 0$ share the same solution y if $k = 6$.

Note, $\mathrm{rank}\Lambda_s \leq 6$ for $k \leq 6$. $\mathrm{rank}\Lambda_s = k \quad (1 \leq k \leq 6)$, $\mathrm{rank}\Lambda_s = 6 \quad (k > 6)$, if $\lambda_i \neq \lambda_j \neq 1 \neq 0$, $(6 \leq i < j \leq 10)$.

Let y_1 $\arg\min_{y_1 \in \mathcal{R}^k} \|\widetilde{\Lambda_s} y_1 - [1, \cdots, 1]^\mathsf{T}\|_2$

$$\begin{aligned}
\min_y \|\Lambda_\epsilon y - [1, \cdots, 1]^\mathsf{T}\|_2 &\leq \|\Lambda_\epsilon y_1 - [1, \cdots, 1]^\mathsf{T}\|_2 && (4.68)\\
&\approx \|\Lambda_s y_1 - [1, \cdots, 1]^\mathsf{T} + P y_1\|_2 && (4.69)\\
&\leq \|\Lambda_s y_1 - [1, \cdots, 1]^\mathsf{T}\|_2 + \|P y_1\|_2 && (4.70)\\
&= \|\widetilde{\Lambda_s} y_1 - [1, \cdots, 1]^\mathsf{T}\|_2 + \|P y_1\|_2 \quad (k = 6) && (4.71)\\
&= \|P y_1\|_2 \quad (k = 6) && (4.72)
\end{aligned}$$

Notice, when $k = 6$, $\|\widetilde{\Lambda_s} y_1 - [1, \cdots, 1]^\mathsf{T}\|_2 = 0$.

$$\|\Lambda_\epsilon y - [1, \cdots, 1]^\mathsf{T}\|_2 \leq \|P y_1\|_2, \tag{4.73}$$

where $y_1 = (y_1^1, y_1^2, \ldots, y_1^6)^\mathsf{T}$, and

$$Py_1 = \begin{pmatrix} \epsilon_1 y_1^1 + 2\epsilon_1 y_1^2 + \cdots + 6\epsilon_1 y_1^6 \\ \epsilon_2 y_1^1 + 2\epsilon_2 y_1^2 + \cdots + 6\epsilon_2 y_1^6 \\ \epsilon_3 y_1^1 + 2\epsilon_3 y_1^2 + \cdots + 6\epsilon_3 y_1^6 \\ \epsilon_4 y_1^1 + 2\epsilon_1 y_4^2 + \cdots + 6\epsilon_4 y_1^6 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{4.74}$$

Since

$$\|\widetilde{\Lambda_s} y_1 - [1, \cdots, 1]^\mathsf{T}\|_2 = 0. \tag{4.75}$$

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_6 & \lambda_6^2 & \cdots & \lambda_6^6 \\ \lambda_7 & \lambda_7^2 & \cdots & \lambda_7^6 \\ \lambda_8 & \lambda_8^2 & \cdots & \lambda_8^6 \\ \lambda_9 & \lambda_9^2 & \cdots & \lambda_9^6 \\ \lambda_{10} & \lambda_{10}^2 & \cdots & \lambda_{10}^6 \end{pmatrix} \begin{pmatrix} y_1^1 \\ y_1^2 \\ y_1^3 \\ y_1^4 \\ y_1^5 \\ y_1^6 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{4.76}$$

which means $1, \lambda_6, \lambda_7, \lambda_8, \lambda_9$ and λ_{10} are roots of

$$f(c) = y_1^6 c^6 + y_1^5 c^5 + y_1^4 c^4 + y_1^3 c^3 + y_1^2 c^2 + y_1^1 c - 1 = 0. \tag{4.77}$$

Thus,

$$f(c) = -\frac{1}{\lambda_6\lambda_7\lambda_8\lambda_9\lambda_{10}}(c-1)(c-\lambda_6)(c-\lambda_7)(c-\lambda_8)(c-\lambda_9)(c-\lambda_{10}). \tag{4.78}$$

$$f'(1) = -\frac{1}{\lambda_6\lambda_7\lambda_8\lambda_9\lambda_{10}}(1-\lambda_6)(1-\lambda_7)(1-\lambda_8)(1-\lambda_9)(1-\lambda_{10}). \tag{4.79}$$

Also,

$$f'(c) = 6y_1^6 c^5 + 5y_1^5 c^4 + 4y_1^4 c^4 + 3y_1^3 c^2 + 2y_1^2 c1 + y_1^1. \tag{4.80}$$

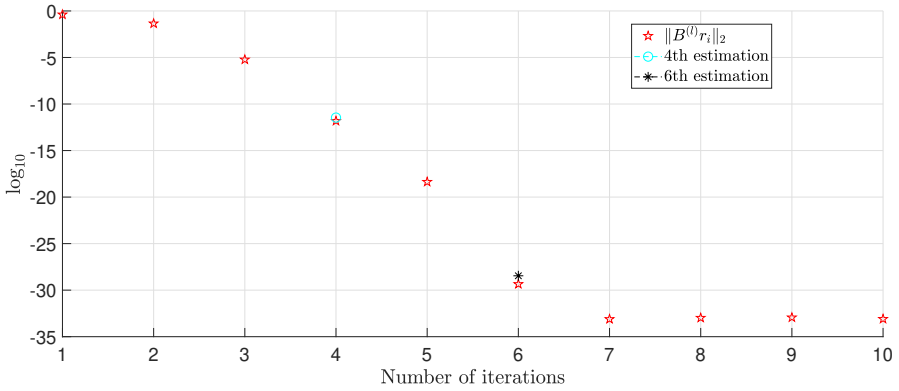


FIGURE 4.6: $\|B^{(l)}r_s\|_2 (l = 8)$ versus the number of iterations for the test matrix A in quadruple precision arithmetic.

$$f'(1) = 6y_1^6 + 5y_1^5 + 4y_1^4 + 3y_1^3 + 2y_1 + y_1^1. \tag{4.81}$$

Let $\epsilon = \max\{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4\} < 10^{-10}$. Note that

$$\|Py_1\|_2 = \|(f'(1)\epsilon_1, f'(1)\epsilon_2, f'(1)\epsilon_3, f'(1)\epsilon_4)^{\mathsf{T}}\|_2 \tag{4.82}$$

$$\leq \epsilon\|(f'(1), f'(1), f'(1), f'(1))^{\mathsf{T}}\|_2 \tag{4.83}$$

$$= 2\epsilon|f'(1)|. \tag{4.84}$$

Since $\|V\|_2 = 2.5068$, we reach the final estimate at the 6th iteration

$$\|B^{(l)}r_s\|_2 = \|B^{(l)}(b - Ax_s)\|_2 \tag{4.85}$$

$$\leq \|V\|_2\|\Lambda_\epsilon y - [1, \cdots, 1]^{\mathsf{T}}\|_2 \tag{4.86}$$

$$\simeq \|V\|_2\|Py_1\|_2 \tag{4.87}$$

$$< \|V\|_2 2\epsilon|f'(1)| \tag{4.88}$$

$$= 2\|V\|_2\epsilon| - \frac{1}{\lambda_6\lambda_7\lambda_8\lambda_9\lambda_{10}}(1-\lambda_6)(1-\lambda_7)(1-\lambda_8)(1-\lambda_9)(1-\lambda_{10})| \tag{4.89}$$

$$\leq 5.136 \times 10^{-10} \times \frac{1}{0.4754} \times 0.4901 \times 0.0675 \times 0.0001 \times (1.8999 \times 10^{-7})^2 \tag{4.90}$$

$$= 3.49 \times 10^{-29}. \tag{4.91}$$

If we choose λ_6 and λ_7 to be in the cluster around 1 , then $\epsilon < 10^{-6}$, and at the 4th iteration we obtain $\|B^{(l)}r_s\|_2 < 3.49 \times 10^{-12}$.

Figure 4.6 shows $\|B^{(l)}r_s\|_2$ versus the number of iterations in quadruple precision

arithmetic (double precisin arithmetic limits the observation). At the 4th iteration $\|B^{(l)}r_s\|_2$ is approximately 10^{-12}, and at 6th iteration $\|B^{(l)}r_s\|_2$ is approximately 10^{-29}, which is close to the estimation. Thus, although A has 10 different singular values, the eigenvalue of the preconditioned matrix B^lA is contained in a cluster around 1. Within several steps, $\|B^{(l)}r_s\|_2$ converges to a tiny level. In other words, it converged near zero before the grade d.

Ipsen's upper bounds for non-normal matrix $B^{(l)}A$ in [47] gives $\|B^{(l)}r_6\|_2 < c\epsilon\|r_0\|_2$, where c is a constant that reflects the distance from separate eigenvalues to the cluster center 1 which is smaller than 0.5 and also reflects the non-normality of $B^{(l)}A$ which is realted to $\|V\|_2$, and $\|r_0\|_2 = 4.55$. Thus, the value of this bound is about 10^{-1}. Ipsen's estimation for normal martrix in [46] is $\|B^{(l)}r_6\|_2 \approx (1/3) \times 0.7^5\|r_0\|_2$, which is larger than our estimation, but $B^{(l)}A$ is non-normal. Our work can be regarded as extending this estimation to the diagonalizable case. Traditional bounds after being log is a straight line, our paper is devoted to illustrating the super linear convergence.

4.3 General proof of the convergence

In the previous section, we analyzed with a specific test matrix A. In this section, we assume that $\widetilde{A} = P^lA^\mathsf{T}A$ is diagonalizable and give a similar estimation for the case when there are more than one cluster, each eigenvalue belongs to a cluster around a center with a small radius.

If there are s cluster centers at step k,

$$\Lambda_\epsilon = \begin{pmatrix} c_1+\epsilon_1 & (c_1+\epsilon_1)^2 & \cdots & (c_1+\epsilon_1)^k \\ c_1+\epsilon_2 & (c_1+\epsilon_2)^2 & \cdots & (c_1+\epsilon_2)^k \\ \cdots & \cdots & \cdots & \cdots \\ c_2+\epsilon_i & (c_2+\epsilon_i)^2 & \cdots & (c_2+\epsilon_i)^k \\ \cdots & \cdots & \cdots & \cdots \\ c_s+\epsilon_d & (c_s+\epsilon_d)^2 & \cdots & (c_s+\epsilon_d)^k \end{pmatrix} \in \mathbb{R}^{d\times k}. \tag{4.92}$$

When ϵ is very small,

$$\widetilde{\Lambda_\epsilon} = \begin{pmatrix} c_1+\epsilon_1 & c_1^2+2c_1\epsilon_1 & \cdots & c_1^k+kc_1^{k-1}\epsilon_1 \\ c_1+\epsilon_2 & c_1^2+2c_1\epsilon_2 & \cdots & c_1^k+kc_1^{k-1}\epsilon_2 \\ \cdots & \cdots & \cdots & \cdots \\ c_2+\epsilon_i & c_2^2+2c_2\epsilon_i & \cdots & c_2^k+kc_2^{k-1}\epsilon_i \\ \cdots & \cdots & \cdots & \cdots \\ c_s+\epsilon_d & c_s^2+2c_s\epsilon_d & \cdots & c_s^k+kc_s^{k-1}\epsilon_d \end{pmatrix} \in \mathbb{R}^{d\times k}. \tag{4.93}$$

$$\Lambda_\epsilon \approx \widetilde{\Lambda_\epsilon} = \Lambda_s + P. \tag{4.94}$$

$$\Lambda_s = \begin{pmatrix} c_1 & c_1^2 & \cdots & c_1^k \\ c_1 & c_1^2 & \cdots & c_1^k \\ \cdots & \cdots & \cdots & \cdots \\ c_2 & c_2^2 & \cdots & c_2^k \\ \cdots & \cdots & \cdots & \cdots \\ c_s & c_s^2 & \cdots & c_s^k \end{pmatrix}, \qquad P = \begin{pmatrix} \epsilon_1 & 2c_1\epsilon_1 & \cdots & kc_1^{k-1}\epsilon_1 \\ \epsilon_2 & 2c_1\epsilon_2 & \cdots & kc_1^{k-1}\epsilon_2 \\ \cdots & \cdots & \cdots & \cdots \\ \epsilon_d & 2c_s\epsilon_d & \cdots & kc_s^{k-1}\epsilon_d \end{pmatrix}. \tag{4.95}$$

Delete the same rows of Λ_s, we obtain $\widetilde{\Lambda_s}$,

$$\widetilde{\Lambda_s} = \begin{pmatrix} c_1 & c_1^2 & \cdots & c_1^k \\ c_2 & c_2^2 & \cdots & c_2^k \\ \cdots & \cdots & \cdots & \cdots \\ c_s & c_s^2 & \cdots & c_s^k \end{pmatrix} \in \mathbb{R}^{s\times k}. \tag{4.96}$$

Let $y_1 = \arg\min_{y\in\mathcal{R}^k} \|\widetilde{\Lambda_s} y - [1,\cdots,1]^\mathsf{T}\|_2$

$$\begin{aligned} \|\Lambda_\epsilon y - [1,\cdots,1]^\mathsf{T}\|_2 &\le \|\Lambda_\epsilon y_1 - [1,\cdots,1]^\mathsf{T}\|_2 && (4.97)\\ &\approx \|\Lambda_s y_1 - [1,\cdots,1]^\mathsf{T} + Py_1\|_2 && (4.98)\\ &\le \|\Lambda_s y_1 - [1,\cdots,1]^\mathsf{T}\|_2 + \|Py_1\|_2 && (4.99)\\ &= \|\widetilde{\Lambda_s} y_1 - [1,\cdots,1]^\mathsf{T}\|_2 + \|Py_1\|_2 \quad (k=s). && (4.100) \end{aligned}$$

$$P = \begin{pmatrix} \epsilon_1 & 2c_1\epsilon_1 & \cdots & kc_1^{k-1}\epsilon_1 \\ \epsilon_2 & 2c_1\epsilon_2 & \cdots & kc_1^{k-1}\epsilon_2 \\ \cdots & \cdots & \cdots & \cdots \\ \epsilon_d & 2c_s\epsilon_d & \cdots & kc_s^{k-1}\epsilon_d \end{pmatrix} \tag{4.101}$$

Let $y_1 = (y_1^1, y_1^2, \cdots, y_1^k)^\mathsf{T}$, then

$$Py_1(c_i) = (ky_1^k c_i^{k-1} + (k-1)y_1^{k-1}c_i^{k-2} + \cdots + y_1^1)\epsilon_i \tag{4.102}$$

$$y_1 = \min_{y \in \mathcal{R}^s} \|\widetilde{\Lambda_s} y - [1, \cdots, 1]^\mathsf{T}\|_2 \tag{4.103}$$

Let $\epsilon = \max\{\epsilon_1, \cdots, \epsilon_d\}$. Define the polynomial $f(\widetilde{c_i})$ by y_1.

$$f(\widetilde{c_i}) = y_1^k \widetilde{c_i}^k + y_1^{k-1}\widetilde{c_i}^{k-1} + \cdots + y_1^1\widetilde{c_i} - 1. \tag{4.104}$$

Thus,

$$f(c) = \frac{1}{\prod_{i=1}^k \widetilde{c_i}} \prod_{i=1}^k (c - \widetilde{c_i}). \tag{4.105}$$

$$f'(\widetilde{c_i}) = \frac{1}{\prod_{i=1}^k \widetilde{c_i}} \prod_{j=1,\cdots,k,j\neq i} (\widetilde{c_i} - \widetilde{c_j}) \tag{4.106}$$

$$f'(c_i) = ky_1^k c_i^{k-1} + (k-1)y_1^{k-1}c_i^{k-2} + \cdots + y_1^1 \tag{4.107}$$

Thus, at step k

$$\|B^{(l)}(b - Ax_k)\|_2 \leq \|V\|_2 \epsilon \|(f'(\widetilde{c_1}), f'(\widetilde{c_2}), \ldots, f'(\widetilde{c_k}))^\mathsf{T}\|_2 \tag{4.108}$$

If $\|V\|_2$ is not large, and all eigenvalues are well clustered, $\|B^{(l)}(r)\|_2$ will converge to a tiny value after s iterations. $\widetilde{c_1}$ should be in the convex hull of c_i, which need to be proved in future. It indicates that as long as c_i are well clustered, $\widetilde{c_1}$ are also well clustered. We can calculate y_1 of the Vandermonde matrix $\widetilde{\Lambda_s}$ and observe that the $\|B^{(l)}r_s\|_2$ and its estimate $\|V\|_2(\|\Lambda_s y_1 - [1, \cdots, 1]^\mathsf{T}\|_2 + \|Py_1\|_2)$ matches well, as shown in Figure 4.7.

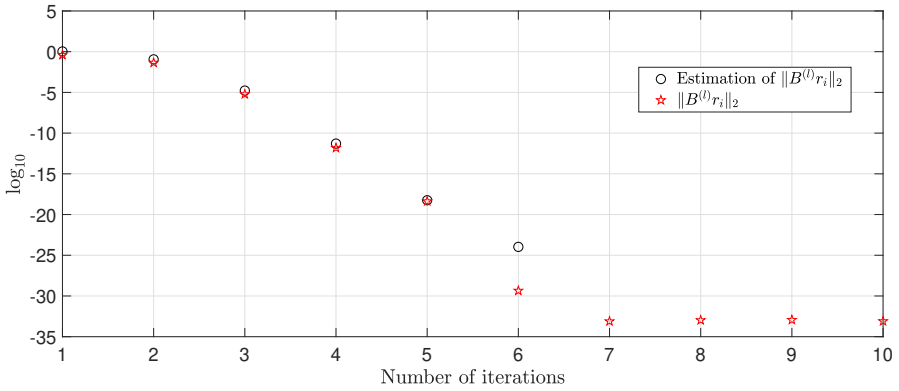


FIGURE 4.7: The $\|B^{(l)}r_s\|_2 (l = 8)$ of the test matrix A in quadruple precision arithmetic.

Chapter 5

Inner-iteration preconditioned block GMRES

This chapter first introduces previous work on the block GMRES method. Then, it reviews the algorithm of the block GMRES method which is based on the block Arnoldi method. Then, it gives the inner-iteration preconditioned block GMRES method and the corresponding numerical experiments. [48, 49]

5.1 Previous work

Consider solving the linear least squares problems with multiple right-hand sides

$$AX = C, \quad A \in \mathbb{R}^{m\times m}, \quad C \in \mathcal{R}^{m\times p}, \tag{5.1}$$

where A is full-rank.

A natural idea is to do the Cholesky factorization, QR factorization, or the singular value decomposition of A, and store the information, and reuse it for each right-hand side. But the factorization process can be heavy, and where A is only available as a function A active as a vecotr x, hard to access a certain element of the matrix, whereas it is easy to get the matrix vector product. Thus, we are more interested in solving the problems with multiple right-hand sides by iterative methods.

The first block method is the block conjugate gradient method which was introduced by O'Leary[50] for symmetric positive definite matrices. For problems with nonsymmetric matrices, a block version of GMRES was developed in [51], which is based on a block Arnoldi process [13]. See also [52]. Many block Arnoldi-type methods differ in the choice of the inner product [51, 53, 54], as studied in [55]. The convergence of block GMRES studied in [56] and [57].

5.2 Inner-iteration preconditioned block GMRES

In this thesis, we are more interested in solving the least squares problems with multiple right-hand sides,

$$\min \|AX - C\|_F, \quad A \in \mathbb{R}^{m\times n}, \quad m > n, \quad C \in \mathbb{R}^{m\times p}. \tag{5.2}$$

which is needed, for instance, in the Cluster Gauss-Newton method [58] and Nonnegative Matrix Factorization [59].

5.2.1 Block Arnoldi method

In this thesis, we use the Block Arnoldi method given in [38]. The algorithm is as below.

Algorithm 10 Block Arnoldi method

1: Choose an unitary matrix V_1 of dimension $n \times p$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Compute $H_{i,j} = V_i^T A V_j, \qquad i = 1, 2, \cdots, j$
4: Compute $W_j = AV_j - \Sigma_{i=1}^{j} V_i H_{i,j}$
5: Compute the QR factorization of $W_j : W_j = V_{j+1} H_{j+1,j}$
6: **end for**

Based on Algorithm 10, we can derive the block GMRES. For $A \in \mathcal{R}^{n\times n}$, let $X_0 \in \mathcal{R}^{n\times p}$ be the initial solution, then, the initial residual $R_0 = C - AX_0 \in \mathcal{R}^{n\times p}$, compute the QR factorization of $R_0 : R_0 = V_1 R$ to get V_1, and approximate solution $X_i = X_0 + Z$, where Z solves

$$\min_{Z\in K_i} \|C - A(X_0 + Z)\|_F = \min_{Z\in K_i} \|R_0 - AZ\|_F$$

$$K_i \equiv \text{span}\{R_0, AR_0, \cdots, A^{i-1}R_0\}.$$

V_i is the orthonormal basis of K_i, which is obtained by the block Arnoldi method.

$$C - A(X + X_0) = R_0 - AX = V_1 R - AV_i Y = V_{i+1}(R - H_{(i+1)p,ip} Y) \tag{5.3}$$

$$\min_Y ||R - H_{(i+1)p,ip} Y||_F \tag{5.4}$$

When $p = 2, i = 3$, H_{86} is a block upper Hessenberg matrix.

$$H_{86} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} & h_{16} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} & h_{26} \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} & h_{36} \\ 0 & h_{42} & h_{43} & h_{44} & h_{45} & h_{46} \\ 0 & 0 & h_{53} & h_{54} & h_{55} & h_{56} \\ 0 & 0 & 0 & h_{64} & h_{65} & h_{66} \\ 0 & 0 & 0 & 0 & h_{75} & h_{76} \\ 0 & 0 & 0 & 0 & 0 & h_{86} \end{pmatrix}, \qquad R = \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{5.5}$$

$$H_{(i+1)p,ip} = Q_{i+1} T_{i+1} \tag{5.6}$$

$$T_{i+1} Y = Q_{i+1}^{\mathsf{T}} R \tag{5.7}$$

$$T_{86} = \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} & t_{15} & t_{16} \\ 0 & t_{22} & t_{23} & t_{24} & t_{25} & t_{26} \\ 0 & 0 & t_{33} & t_{34} & t_{35} & t_{36} \\ 0 & 0 & 0 & t_{44} & t_{45} & t_{46} \\ 0 & 0 & 0 & 0 & t_{55} & t_{56} \\ 0 & 0 & 0 & 0 & 0 & t_{66} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad Q_{i+1}^{\mathsf{T}} R = \begin{pmatrix} * & * \\ * & * \\ * & * \\ * & * \\ * & * \\ * & * \\ r_{71} & r_{72} \\ 0 & r_{82} \end{pmatrix}. \tag{5.8}$$

We can only solve the triangular system and corresponding right-hand sides by backward substitution. The 0 rows of T_{86} correspond to a triangular matrix in the $Q_{i+1}^{\mathsf{T}} R$, which is made of residuals. We estimate the F-norm of this residual matrix.

Similarly to GMRES, T_{m+1} and the corresponding right-hand sides can be updated step by step by storing the rotation matrix in the QR decomposition.

The algorithm of Block BA-GMRES is as follows.

Algorithm 11 Blcok BA-GMRES

Choose $X_0 \in \mathbb{R}^{n \times p}$, $R_0 = C - AX_0$, $[V_1, R] = qr(BR_0)$,
for $i = 1, 2, \ldots, k$ **do**
  $W_i = BAV_i$,
  for $j = 1, 2, \ldots, i$ **do**
    $H_{i,j} = V_j^{\mathsf{T}} W_i$, $W_i = W_i - V_j H_{i,j}$,
  end for
  $[V_{i+1}, H_{i+1,i}] = qr(W_i)$,
  Compute $Y_i \in \mathbb{R}^{i \times p}$ which minimizes $\|R_i\|_F = \|R - H_{(i+1)p,ip} Y_i\|_F$,
  $X_i = X_0 + [V_1, V_2, \ldots, V_i] Y_i$, $R_i = C - AX_i$.
  if $\|A^{\mathsf{T}} R_i\|_F < \epsilon \|A^{\mathsf{T}} R_0\|_F$ **then**
    stop
  end if
end for

5.3 Inner-iteration block BA-GMRES

In order to get inner-iteration block BA-GMRES, we need to adapt the stationary methods to block algorithms at first. We still use NR-SOR as the main method. The block NR-SOR is developed from algorithm 8 (NR-SOR), the algorithm is given as follows.

Algorithm 12 Block NR-SOR

Let X^0 be the initial solution and $R = C - AX^0$, $0 < \omega < 2$.
for $k = 1, 2, \ldots, l$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    $\Delta_i^\mathsf{T} = (\omega/||a_i||_2^2)R^\mathsf{T}a_i$,
    $X_i^{k+1\mathsf{T}} = X_i^{k\mathsf{T}} + \Delta_i^\mathsf{T}$,
    $R = R - a_i\Delta_i^\mathsf{T}$ (rank-1 update).
  end for
end for

Consider the case $B = A^\mathsf{T}$ in block BA-GMRES, and combine the block BA-GMRES with block NR-SOR. Then, the algorithm is as follows.

Algorithm 13 NR-SOR inner-iteration block BA-GMRES

Choose $X_0 \in \mathbb{R}^{n\times p}$, $R_0 = C - AX_0$,
apply l steps SOR to $A^\mathsf{T}Aw = A^\mathsf{T}R_0$ to obtain $W_0 = P^lA^\mathsf{T}R_0$, (NR-SOR),
$[V_1, R] = qr(W_0)$,
for $i = 1, 2, \ldots, k$ **do**
  $U_i = AV_i$,
  apply l steps SOR to $A^\mathsf{T}AW = A^\mathsf{T}U_i$ to obtain $W_i = P^lA^\mathsf{T}U_i$, (NR-SOR)
  for $j = 1, 2, \ldots, i$ **do**
    $H_{i,j} = V_j^\mathsf{T}W_i$, $W_i = W_i - V_jH_{i,j}$,
  end for
  $[V_{i+1}, H_{i+1,i}] = qr(W_i)$,
  Compute $Y_i \in \mathbb{R}^{i\times p}$ which minimizes $\|R_i\|_F = \|R - H_{(i+1)p,ip}Y_i\|_F$,
  $X_i = X_0 + [V_1, V_2, \ldots, V_i]Y_i$, $R_i = C - AX_i$.
  if $\|A^\mathsf{T}R_i\|_F < \epsilon\|A^\mathsf{T}R_0\|_F$ **then**
    stop
  end if
end for

5.4 Numerical experiments

5.4.1 Numerical experiments of block BA-GMRES

Figure 5.1 shows the experiment with the matrix Maragal3 given in Table 3.1, whose size is 1682×858 and the rank is 613. The BA-GMRES needs 547 steps to converge for

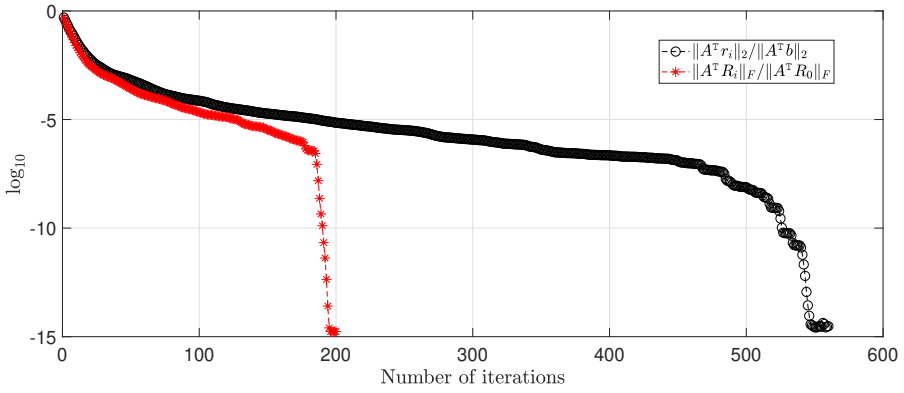


FIGURE 5.1: The comparison between BA-GMRES and block BA-GMRES (p=3, $B = A^T$).

TABLE 5.1: Iteration steps and CPU time of Block BA-GMRES

p	iteration steps	CPU time (s)	$\|A^T R_i\|_F/\|A^T R_0\|_F$
1	547	0.4075	8.91×10^{-15}
2	285	0.6229	9.27×10^{-15}
3	195	0.3908	6.65×10^{-15}
4	149	0.3268	6.27×10^{-15}
5	121	0.2722	6.16×10^{-15}
6	102	0.2344	6.18×10^{-15}
7	88	0.2307	5.36×10^{-15}

1 right-hand side. The block BA-GMRES needs 195 steps to converge for 3 right-hand sides, slightly larger than 1/3 steps of BA-GMRES. The relative residual norm of both methods reached a level of 10^{-14}. Thus, we test more cases and compute the CPU time, the stopping criterion is $\epsilon = 10^{-14}$, where p represents the number of the right-hand sides.

Table 5.1 shows the numerical results of the block BA-GMRES for $p = 2, 3, \ldots, 7$, where $p = 1$ is the BA-GMRES. The relative residual norm $\|A^T R_i\|_F/\|A^T R_0\|_F$ reached a similar level 10^{-14}, and did not become larger for more right-hand sides cases. The CPU time decreased for more right-hand sides. For $p = 7$, the CPU is only $0.2307s$ which means solving 7 problems at the same time is faster than solving 1 problem which costs $0.4075s$. The first reason is the effect of the block algorithm, where the memory access is more efficient compared to the non-block algorithm. The second reason is the iteration steps is decreasing with the increase of the number of right-hand sides. The numerical results showed that the block BA-GMRES is efficient.

TABLE 5.2: CPU time of block BA-GMRES and IP Block BA-GMRES

p	Iter.(B=A^T)	CPU (s)	Iter.(IP)	CPU (s)
1	547	0.4075	201	0.2535
2	285	0.6229	126	0.2025
3	195	0.3908	92	0.1707
4	149	0.3268	76	0.1751
5	121	0.2722	65	0.1728
6	102	0.2344	58	0.1728
7	88	0.2307	53	0.1842

5.4.2 Numerical experiments of NR-SOR inner-iteration preconditioned block BA-GMRES

We also tested Maragal3 with the NR-SOR inner-iteration block BA-GMRES, where the number of inner-iteration was set to 1. In MATLAB, doing more inner-iteration steps is time consuming, but for 1 inner iteration it can be implemented by backward substitution, which is equivalent to the backslash for a triangular system. The result is as below, where IP stands for inner-iteration preconditioned block BA-GMERES (NR-SOR).

Table 5.2 shows the comparison of the CPU of the block BA-GMRES and the block IP BA-GMRES. As for iteration steps, the block IP BA-GMRES is nearly half of the block BA-GMRES. But the inner-iteration process costs time, so that the CPU time is not half. The CPU time decreased after using the inner-iteration preconditioning. The best result is for $p = 6$, which means solving 6 problems at the same time only needs 0.1728s. The numerical result shows the proposed method is more effective than the block BA-GMRES.

Chapter 6

Conclusion and future work

6.1 Concluding Remarks

We proposed a stabilized AB-GMRES method for ill-conditioned underdetermined and inconsistent least squares problems. It shifts upwards the tiny singular values of the upper triangular matrix appearing in AB-GMRES, making the process more stable, giving better convergence, and more accurate solutions compared to AB-GMRES. The method is also effective for making AB-GMRES stable for inconsistent least squares problems with highly ill-conditioned square coefficient matrices.

Next, we analyzed the convergence of inner-iteration preconditioned GMRES method for overdetermined least squares problems based on the distribution of the eigenvalues of the preconditioned matrix. One can choose some eigenvalue λ_i as a center, and choose ϵ_i as the radius, where many eigenvalues λ lie in $|\lambda - \lambda_i| < \epsilon_i$. Let $\epsilon = \max \epsilon_i$. Assume there are $j+1$ clusters of the eigenvalues of $B^{(l)}A$. If $B^{(l)}A$ is diagonalizable, we can prove that at step $j+1$, the upper bound of the residual $\|B^{(l)}r_{j+1}\|$ is $\mathbf{O}(\prod_1^j(1-\lambda_j)\epsilon)$. This explains why the inner-iteration preconditioning enables the convergence in a relatively small number of steps. Due to the clustering of the eigenvalues, at $j+1$ steps, $\|B^{(l)}r_{j+1}\|$ can converge to a tiny value.

Finally, we proposed the block IP-GMRES method, which combines the inner-iteration preconditioning technique and the block GREMS method. The proposed method reduces the iteration steps by nearly a half and also reduces the CPU time compared to the block GMRES method, which means the proposed method is effective.

6.2 Future work

For the stabilized method, the $\mathrm{LDL}^{\mathsf{T}}$ decomposition needs to be compared with the QR decomposition. For the convergence analysis, the non diagonalizable (Jordan block) case

needs to be analyzed. The convergence for cases with multiple clusters also need more research. For the IP block-GMRES part, coding in C and analyzing the convergence versus the number of inner-iteration steps is needed. The grade of block

$$\text{grade}(A, C) = \min\{d | K_d(A, C) = K_{d+1}(A, C)\} \tag{6.1}$$

is also interesting, and the convergence analysis of IP block GMRES will be investigated.

Appendix A

Previous work

A.1 Comparisons with other methods

A.1.1 Underdetermined inconsistent least squares problems

First, we compared the stabilized AB-GMRES with the range restricted AB-GMRES (RR-AB-GMRES) [33], where the Krylov subspace for the RR-AB-GMRES with $B = A^\mathsf{T}$ is $K_i(AA^\mathsf{T}, AA^\mathsf{T}r_0)$, AB-GMRES with $B = A^\mathsf{T}$, BA-GMRES with $B = A^\mathsf{T}$, LSQR [5] and LSMR [6]. All programs for iterative methods were coded according to the algorithms in [5, 6, 8, 33]. Each method was terminated at the iteration step which gives the minimum relative residual norm within m iterations, where m is the number of the rows of the matrix. No restarts were used for GMRES. Experiments were done for rank-deficient matrices whose information is given in Table 1. Here, we have deleted the zero rows and columns of the test matrices beforehand. The elements of b were randomly generated using the MATLAB function `rand`. Each experiment was done 10 times for the same right hand side b and the average of the CPU times are shown. Symbol - denotes that $\|A^\mathsf{T}r_i\|_2/\|A^\mathsf{T}r_0\|_2$ did not reach 10^{-8} within $20n$ iterations.

Table A.1 shows that the stabilized AB-GMRES is generally more accurate than the RR-AB-GMRES. The stabilized AB-GMRES took more iterations to attain the same order of the smallest residual norm than the RR-AB-GMRES. Table A.1 also shows that for the same underdetermined least squares problems, the BA-GMRES was the best in terms of the attainable smallest relative residual norm and that the LSQR and LSMR are comparable to the BA-GMRES, but require less CPU time according to Tabel A.2.

TABLE A.1: Comparison of the attainable smallest relative residual norm $\|A^{\mathsf{T}}r_i\|_2/\|A^{\mathsf{T}}r_0\|_2$.

matrix	Maragal_3T	Maragal_4T	Maragal_5T	Maragal_6T	Maragal_7T
iter.	531	465	1110	2440	1864
standard AB-GMRES	1.05×10^{-8}	2.09×10^{-7}	5.35×10^{-6}	8.26×10^{-6}	4.53×10^{-6}
iter.	552	598	1226	3002	2459
stabilized AB-GMRES	5.99×10^{-12}	5.59×10^{-8}	4.22×10^{-6}	3.88×10^{-6}	2.80×10^{-7}
iter.	553	565	1223	2374	2474
RR-AB-GMRES	2.57×10^{-11}	5.59×10^{-8}	3.62×10^{-6}	1.63×10^{-5}	2.78×10^{-7}
iter.	562	626	1263	4373	5658
BA-GMRES	2.88×10^{-14}	7.92×10^{-11}	2.29×10^{-12}	5.12×10^{-11}	2.03×10^{-10}
iter.	1682	2375	4576	151013	97348
LSQR	5.64×10^{-14}	2.77×10^{-10}	1.11×10^{-11}	5.87×10^{-10}	1.33×10^{-9}
iter.	1654	2308	4273	127450	70242
LSMR	5.51×10^{-14}	3.00×10^{-10}	3.25×10^{-11}	4.16×10^{-10}	9.95×10^{-10}

TABLE A.2: Comparison of the CPU time (seconds) to obtain relative residual norm $\|A^{\mathsf{T}}r_i\|_2/\|A^{\mathsf{T}}r_0\|_2 < 10^{-8}$.

matrix	Maragal_3T	Maragal_4T	Maragal_5T	Maragal_6T	Maragal_7T
iter.	-	-	-	-	-
standard AB-GMRES	-	-	-	-	-
iter.	546 (526)	-	-	-	-
stabilized AB-GMRES	2.01	-	-	-	-
iter.	545	-	-	-	-
RR-AB-GMRES	1.84	-	-	-	-
iter.	530	608	1232	3623	5001
BA-GMRES	2.10	3.19	4.25×10^{1}	1.81×10^{3}	9.20×10^{3}
iter.	1465	2120	4032	101893	54444
LSQR	1.27×10^{-1}	2.56×10^{-1}	1.49	2.93×10^{2}	4.33×10^{2}
iter.	1456	1989	4013	54017	31206
LSMR	1.25×10^{-1}	2.37×10^{-1}	1.49	1.50×10^{2}	2.23×10^{2}

TABLE A.3: Information of the singular square matrices.

matrix	size	density[%]	rank	$\kappa_2(A)$	application
Harvard500	500	1.05	170	1.30×10^2	web connectivity
netz4504	1961	0.13	1342	3.41×10^1	2D/3D finite element problem
TS	2142	0.99	2140	3.52×10^3	counter example problem
grid2_dual	3136	0.12	3134	8.58×10^3	2D/3D finite element problem
uk	4828	0.06	4814	6.62×10^3	undirected graph
bw42	10000	0.05	9999	2.03×10^3	partial differential equation [19]

A.1.2 Inconsistent systems with highly ill-conditioned square coefficient matrices

The stabilized AB-GMRES is not restricted to solving underdetermined problems but can also be applied to solving the least squares problem $\min_{x\in\mathbb{R}^n}\|b-Ax\|_2$, where $A\in\mathbb{R}^{n\times n}$ is a highly ill-conditioned square matrix. Thus, we also test on square matrices of different kinds. Table A.3 gives the information of the matrices.

These matrices are all numerically singular. We generated the right-hand side b by the MATLAB function `rand`, so that the systems are generically inconsistent. We compared the stabilized AB-GMRES with the standard AB-GMRES, RR-AB-GMRES, BA-GMRES with $B=A^\mathsf{T}$, LSMR [6], and LSQR [5]. Table A.4 gives the smallest relative residual norm and the number of iterations. Table A.6 gives the CPU times in seconds required to obtain relative residual norm $\|A^\mathsf{T}r_i\|_2/\|A^\mathsf{T}r_0\|_2<10^{-8}$. The switching strategy which was introduced in Section 3.4.1 was used for the stabilized AB-GMRES when measuing CPU times. The number of iterations when switching occurred is in brackets.

Table A.4 shows that for most problems the BA-GMRES was the best in terms of accuracy of relative residual norm. The LSQR and LSMR are similar and are comparable to the BA-GMRES, beacuse they all change the inconsistent problem into a consistent problem. The LSQR and LSMR are more suitable for large and sparse problems compared to the BA-GMRES because they require less CPU time and memory.

For Harvard500 and bw42, the AB-GMRES could only converge to the level of 10^{-9} regarding the relative residual norm, while the stabilized AB-GMRES converged to the level of 10^{-14}. The stabilized AB-GMRES was robust in the sense that it could continue to compute even when the upper triangular matrix R_i became seriously ill-conditioned, and the relative residual norm did not increase sharply towards the end, but just stagnated at a low level, just like for consistent problems. Comparing the CPU time in Tabel A.6, LSMR was the fastest. The stabilized AB-GMRES was usually faster than BA-GMRES.

TABLE A.4: Comparison of the attainable smallest relative residual norm $\|A^{\mathsf{T}}r_i\|_2/\|A^{\mathsf{T}}r_0\|_2$ for inconsistent square linear systems.

matrix	Harvard500	netz4504	TS	grid2_dual	uk	bw42
iter. standard AB-GMRES	104 9.38×10^{-9}	144 4.51×10^{-10}	1487 1.56×10^{-9}	3134 5.98×10^{-10}	4620 1.35×10^{-9}	715 8.06×10^{-8}
iter. stabilized AB-GMRES	175 4.53×10^{-14}	201 1.51×10^{-14}	1617 1.54×10^{-9}	3135 1.14×10^{-9}	4779 6.81×10^{-10}	788 1.66×10^{-7}
iter. RR-AB-GMRES	135 7.78×10^{-14}	200 3.36×10^{-14}	1652 4.56×10^{-9}	3134 6.52×10^{-8}	4706 8.33×10^{-8}	1163 1.56×10^{-5}
iter. BA-GMRES	139 1.91×10^{-15}	194 7.27×10^{-16}	1628 8.43×10^{-13}	3134 1.23×10^{-13}	4724 6.94×10^{-14}	1520 1.97×10^{-11}
iter. LSQR	391 3.59×10^{-15}	198 5.86×10^{-16}	6047 1.96×10^{-12}	12549 2.51×10^{-13}	6249 6.56×10^{-14}	1256 1.59×10^{-11}
iter. LSMR	338 2.01×10^{-15}	195 5.97×10^{-16}	6219 1.25×10^{-12}	12497 2.34×10^{-13}	6199 7.35×10^{-14}	1212 1.60×10^{-11}

TABLE A.5: Attainable smallest relative residual norm $\|A^{\mathsf{T}}r_i\|_2/\|A^{\mathsf{T}}r_0\|_2$ for bw42.

method	iter.	$\min_i \|A^{\mathsf{T}}r_i\|_2/\|A^{\mathsf{T}}r_0\|_2$
standard GMRES	147	8.08×10^{-9}
stabilized GMRES	219	1.94×10^{-11}
RR-GMRES	220	3.13×10^{-11}

TABLE A.6: Comparison of the CPU time (seconds) to obtain relative residual norm $\|A^{\mathsf{T}}r_i\|_2/\|A^{\mathsf{T}}r_0\|_2 < 10^{-8}$ for inconsistent square linear systems.

matrix	Harvard500	netz4504	TS	grid2_dual	uk	bw42
iter. standard AB-GMRES	104 4.72×10^{-2}	134 1.87×10^{-1}	1411 2.14×10	3134 2.16×10^{2}	4583 6.93×10^{2}	- -
iter. stabilized AB-GMRES	104 4.78×10^{-2}	134 1.89×10^{-1}	1531 (182) 8.19×10	3134 2.21×10^{2}	4679 (4199) 1.93×10^{3}	- -
iter. RR-AB-GMRES	114 6.42×10^{-2}	153 2.62×10^{-1}	1530 2.68×10	- -	- -	- -
iter. BA-GMRES	103 5.48×10^{-2}	131 1.72×10^{-1}	1379 2.06×10	3134 2.44×10^{2}	4562 7.55×10^{2}	738 2.33×10
iter. LSQR	222 5.63×10^{-3}	134 6.61×10^{-3}	4239 7.86×10^{-1}	11802 1.15	5948 8.65×10^{-1}	913 3.12×10^{-1}
iter. LSMR	215 5.34×10^{-3}	132 6.42×10^{-3}	3913 7.04×10^{-1}	11746 1.15	5898 8.42×10^{-1}	655 2.32×10^{-1}

Thus, our stabilization method also makes AB-GMRES stable for highly ill-conditioned inconsistent systems with square coefficient matrices.

The coefficient matrix A of bw42 is singular and satisfies $\mathcal{N}(A) = \mathcal{N}(A^{\mathsf{T}})$. The problem comes from a finite-difference discritization of a PDE with periodic boundary condition (Experiment 4.2 in Brown and Walker [19] with the original b). Since the matrix is range symmetric, the GMRES, RR-GMRES, and stabilized GMRES can be directly applied to $Ax = b$ (See [19] Theorem 2.4, [35] Theorem 2.7, and [20] Theorem 3.2.) as shown in Table A.5. The stabilized GMRES gave the relative residual norm 1.94×10^{-11} for bw42 at the 219th iteration, similar to the BA-GMRES.

Appendix B

Grade of Block GMRES

For the multiple right hand sides problem,

$$AX = C, \qquad C = [c_1, \cdots, c_p]. \tag{B.1}$$

The span of the block C is

$$< C >= \text{span}\{c_1, \cdots, c_p\} \tag{B.2}$$

The grade for the block C can be defined by the block Krylov subspace

$$K_d(A, C) =< C > + < AC > +...+ < A^{d-1}C >, \tag{B.3}$$

as follows:

$$grade(A, C) = \min\{d | K_d(A, C) = K_{d+1}(A, C)\}. \tag{B.4}$$

For a random diagonalizable matrix $A \in \mathbb{R}^{10\times 10}$, let $v_k, k = 1, 2, \ldots, 10$, be the kth eigenvector. The test results are collected in Table B.1. The Grade of a block C is influenced by the structure of the block, like the eigenvectors contained in the block, and whether the $c_i, i = 1, 2, \ldots, p$ are independent or not. The grade of the block has an upper bound, for a random case it is supposed to be $\frac{n}{p}$, where n is the number of essentially different eigenvectors.

c_1 contains four eigenvectors, thus, it needs four iteration steps to reach A-invariance. For the same reason, c_2 needs three. Because c_1 and c_2 are linearly independent, the combination of c_1 and c_2 needs the maximum of them, four steps to become A-invariant. But c_1 and v_1, only need three, since v_1, c_1, Av_1, Ac_1, A^2v_1 and A^2c_1 can make up the space given by c_1, Ac_1, A^2c_1 and A^3c_1. If c_1 combine with the $v_1 - v_2$, you only need two steps. Notice that the maximum grade among cases which contain two right-hand sides is five, but it can be less than five in special cases. Generically, each right-hand side contains all eigenvectors. Thus, the grade is usually $\left\lceil \frac{n}{p} \right\rceil$.

TABLE B.1: Grade of block with different structure of c_i.

p	C	Iteration steps (Grade for reaching invariant)
1	$c_1 = v_1 + v_2 + v_3 + v_4$	4
1	$c_2 = v_8 + v_9 + v_{10}$	3
2	$[c_1 \quad c_2]$	4
2	$[c_1 \quad v_1]$	3
2	$[c_1 \quad v_1 - v_2]$	2
2	$[c_1 \quad v_1 - v_2 + v_3]$	2
2	$[c_1 + c_2 \quad v_1]$	5

TABLE B.2: Case when eigenvalue 1 has multiplicity 10.

p	Iteration steps (Grade for reaching invariant)
1	25
2	13
3	9
4	7
8	4
12	3
24	2

TABLE B.3: Case when eigenvalues 1 and 2 have multiplicity 10.

p	Iteration steps (Grade for reaching invariant)
1	24
2	13
3	10
4	8
8	5
12	4
24	3

We test examples for a diagonal matrix $A \in \mathbb{R}^{34\times 34}$, where the eigenvalue 1 has multiplicity 10, other simple eigenvalues are $2, 3, \ldots, 25$. We generate right-hand sides randomly.

From Table B.2, the eigenvalue 1 has multiplicity. We conjecture that the grade is approximately $\left\lceil \frac{n-1}{p} \right\rceil + 1$.

For a diagonal matrix $A \in \mathbb{R}^{42\times 42}$, where eigenvalues 1 and 2 have multiplicity 10, other simple eigenvalues are $3, 4, \ldots, 24$. We generate right-hand sides randomly.

From Table B.3, where two eigenvalues have multiplicity 10, the grade is supposed to be $\left\lceil \frac{n-2}{p} \right\rceil + 2$, where n is the number of eigenvalues.

We conjecture that, if A contains i eigenvalues which have multiplicities, the grade is $\left\lceil \frac{n-i}{p} \right\rceil + i$, where p is the number of right-hand sides in the block, n is the number of

different eigenvalues and i is the number of eigenvalues which have multiplicities. This suggests that you may need extra iterations for the eigenvalues which have multiplicities.

Bibliography

[1] Kaczmarz S. Angenäherte Auflösung von Systemen linearer Glei-chungen. Bull Int Acad Pol Sic Let, Cl Sci Math Nat. 1937:355-7.

[2] Cimmino G. Calcolo approssimato per le soluzioni dei systemi di equazioni lineari. La Ricerca Scientifica (Roma). 1938;1:326-33.

[3] Young DM. Convergence properties of the symmetric and unsymmetric successive overrelaxation methods and related methods. Math Comput. 1970;24(112):793-807.

[4] Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. J Research Nat Bur Standards. 1952;49(6):409-36.

[5] Paige CC, Saunders MA. LSQR: An algorithm for sparse linear equations and sparse least squares. ACM Trans Math Software. 1982;8(1):43-71.

[6] Fong DCL, Saunders M. LSMR: An iterative algorithm for sparse least-squares problems. SIAM J Sci Comput. 2011;33(5):2950-71.

[7] Saad Y, Schultz MH. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput. 1986;7(3):856-69.

[8] Hayami K, Yin JF, Ito T. GMRES methods for least squares problems. SIAM J Matrix Anal Appl. 2010;31(5):2400-30.

[9] Morikuni K. Inner-iteration Preconditioning for Least Squares Problems. Doctoral Thesis, Department of Informatics, School of Multidisciplinary Sciences, The Graduate University for Advanced Studies; 2013.

[10] Morikuni K, Hayami K. Convergence of inner-iteration GMRES methods for rank-deficient least squares problems. SIAM J Matrix Anal Appl. 2015;36(1):225-50.

[11] ADVANPIX LLC. Multiprecision Computing Toolbox for MATLAB;. Version 4.4.5.12711. Available from: https://www.advanpix.com/.

[12] Paige CC, Saunders MA. Solution of sparse indefinite systems of linear equations. SIAM J Numer Anal. 1975;12(4):617-29.

[13] Arnoldi WE. The principle of minimized iterations in the solution of the matrix eigenvalue problem. Q APPL MATH. 1951;9(1):17-29.

[14] Liao Z, Hayami K, Yin JF. A Stabilized GMRES Method for Solving Inconsistent Underdetermined Least Squares Problems. The 22nd Meeting of the JSIAM special interest Group on Algorithms for Matrix/Eigenvalue Problems and their Applications, The University of Tokyo; November 25th, 2016. .

[15] Liao Z, Hayami K. Stabilized GMRES method using the normal equation approach for highly ill-conditioned problems. The 24th Meeting of the JSIAM special interest Group on Algorithms for Matrix/Eigenvalue Problems and their Applications, The University of Tokyo; November 24th, 2017. .

[16] Liao Z, Hayami K, Morikuni K. Stabilizing GMRES using the normal equation approach for severely ill-conditioned problems. SIAM Conference on Applied Linear Algebra (SIAM-ALA18), Hong Kong; May 4-8th, 2018. .

[17] Liao Z, Hayami K, Morikuni K, Yin JF. A stabilized GMRES method for singular and severely ill-conditioned systems of linear equations. Japan Journal of Industrial and Applied Mathematics. 2022;39(2):717-51.

[18] Björck Å. Numerical Methods for Least Squares Problems. SIAM. Philadelphia, PA; 1996.

[19] Brown P, Walker H. GMRES on (nearly) singular systems. SIAM J Matrix Anal Appl. 1997;18(1):37-51.

[20] Calvetti D, Lewis B, Reichel L. GMRES-type methods for inconsistent systems. Linear Algebra Appl. 2000;316(1-3):157-69.

[21] Reichel L, Ye Q. Breakdown-free GMRES for singular systems. SIAM J Matrix Anal Appl. 2005;26(4):1001-21.

[22] Morikuni K, Rozložník M. On GMRES for singular EP and GP systems. SIAM J Matrix Anal Appl. 2018;39(2):1033-48.

[23] Meza JC, Symes WW. Deflated Krylov subspace methods for nearly singular linear systems. J Optim Theory Appl. 1992;72(3):441-57.

[24] Higham NJ. *Accuracy and Stability of Numerical Algorithms,* Second ed. SIAM. Philadelphia, PA; 2002.

[25] Davis T, Hu Y. The University of Florida sparse matrix collection. ACM Trans Math Software. 2011;38(1):1-25.

[26] Foster L. San Jose State University Singular Matrix Database;. Available from: http://www.math.sjsu.edu/singular/matrices/.

[27] Tebbens JD, Tůma M. On incremental condition estimators in the 2-norm. SIAM J Anal Appl. 2014;35(1):174-97.

[28] Iri M. General Theory of Linear Algebra. Asakura, (in Japanese); 2009.

[29] Horn RA, Johnson CR. Matrix Analysis. Cambridge University Press, New York, NY, 2nd ed.; 2012.

[30] Yamamoto Y, Nakatsukasa Y, Yanagisawa Y, Fukaya T. Roundoff error analysis of the CholeskyQR2 algorithm. Electron Trans Numer Anal. 2015;44(01):306-26.

[31] Sugihara K, Hayami K, Liao Z. GMRES using pseudo-inverse for range symmetric singular systems. arXiv: 2201.11429, 2022.

[32] Higham NJ. The Test Matrix Toolbox for MATLAB (version 3.0). University of Manchester, Manchester; 1995.

[33] Neuman A, Reichel L, Sadok H. Algorithms for range restricted iterative methods for linear discrete ill-posed problems. Numer Algorithms. 2012;59(2):325-31.

[34] Brezinski C, Rodriguez G, Seatzu S. Error estimates for the regularization of least squares problems. Numer Algorithms. 2009;51(1):61-76.

[35] Hayami K, Sugihara M. A geometric view of Krylov subspace methods on singular systems. Numer Linear Algebra Appl. 2011;18(3):449-69.

[36] Liao Z, Hayami K. Convergence analysis of inner-iteration preconditioned GMRES method for least squares problems. 2019 Annual Meeting of the Japan Society for Industrial and Applied Mathematics (JSIAM), The University of Tokyo; September 3-5th, 2019. .

[37] Liao Z, Hayami K. Convergence analysis of inner-iteration preconditioned GMRES method for least squares problems. 16th Joint Meeting of Special Interest Groups, JSIAM, Chuo University, Tokyo; March 4-5th, 2020. .

[38] Saad Y. Iterative Methods for Sparse Linear Systems, 2nd ed. SIAM. Philadelphia, PA; 2003.

[39] De la Garza A. An iterative method for solving systems of linear equations. vol. 731. US Atomic Energy Commission, Technical Information Service; 1951.

[40] Dax A. The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations. SIAM review. 1990;32(4):611-35.

[41] Saad Y. A flexible inner-outer preconditioned GMRES algorithm. SIAM J Sci Comput. 1993;14(2):461-9.

[42] Abe K, Zhang SL. A variable preconditioning using the SOR method for GCR-like methods. Int J Numer Anal Model. 2005;2(2):147-61.

[43] DeLong MA. SOR as a preconditioner. Citeseer; 1997.

[44] DeLong M, Ortega J. SOR as a preconditioner II. Applied numerical mathematics. 1998;26(4):465-82.

[45] Varga RS. Matrix Iterative analysis. Springer; 1962.

[46] Ipsen IC. Expressions and bounds for the GMRES residual. BIT. 2000;40(3):524-35.

[47] Campbell SL, Ipsen IC, Kelley CT, Meyer CD. GMRES and the minimal polynomial. BIT. 1996;36(4):664-75.

[48] Liao Z, Hayami K. Inner-iteration preconditioned block GMRES for least squares problems with multiple right-hand sides. The 26th Meeting of the JSIAM special interest Group on Algorithms for Matrix/Eigenvalue Problems and their Applications, Musashino University, Tokyo; Novermber 28th, 2018. .

[49] Liao Z, Hayami K. Inner-iteration preconditioned block GMRES for least squares problems with multiple right-hand sides. The 9th International Congress on Industrial and Applied Mathematics (ICIAM 2019), University de Valencia; July 15-19th, 2019. .

[50] O'Leary DP. The block conjugate gradient algorithm and related methods. Linear Algebra Appl. 1980;29:293-322.

[51] Vital B. Etude de quelques méthodes de résolution de problemes linéaires de grande taille sur multiprocesseur. Ph.D. thesis, Université de Rennes I, Rennes, France; 1990. (in French).

[52] Saad Y. Numerical methods for large eigenvalue problems. Manchester University Press; 1992.

[53] Simoncini V, Gallopoulos E. An iterative method for nonsymmetric systems with multiple right-hand sides. SIAM J Sci Comput. 1995;16(4):917-33.

[54] Simoncini V, Gallopoulos E. Convergence properties of block GMRES and matrix polynomials. Linear Algebra and its Applications. 1996;247:97-119.

[55] Elbouyahyaoui L, Messaoudi A, Sadok H. Algebraic properties of the block GMRES and block Arnoldi methods. Electronic Transactions on Numerical Analysis. 2009;33(207–220):4.

[56] Drineas P, Ipsen IC, Kontopoulou EM, Magdon-Ismail M. Structural convergence results for approximation of dominant subspaces from block Krylov spaces. SIAM J Matrix Anal A. 2018;39(2):567-86.

[57] Kubínová M, Soodhalter KM. Admissible and attainable convergence behavior of block Arnoldi and GMRES. SIAM J Matrix Anal A. 2020;41(2):464-86.

[58] Aoki Y, Hayami K, Toshimoto K, Sugiyama Y. Cluster Gauss-Newton method for sampling multiple solutions of nonlinear least squares problems-with applications to pharmacokinetic models. arXiv preprint arXiv:180806714. 2018.

[59] Ning Z, Hayami K, Ono N. Fast Solution of Nonnegative Matrix Factorization Via a Matrix-Based Active Set Method. MS95 Matrix Computations with Applications - Part I of II, 18th SIAM Conference on Parallel Processing for Scientific Computing; 2018. .