

Palestine Polytechnic University
Deanship of Graduate Studies and Scientific Research
Master Program of Mathematics



Iterative methods for Moore-Penrose inverse

Zainab Abu-Iram

M.Sc. Thesis

Hebron - Palestine

December, 2018

Iterative methods for Moore-Penrose inverse

Prepared by: Zainab Abu-Iram

Supervisor: Dr. Ali Zein

M.Sc. Thesis

Submitted to the Department of Mathematics at Palestine Polytechnic University as a partial fulfillment of the requirement for the degree of Master of Science.

Hebron - Palestine

Dedication

To my parents,

To my husband Taleb and his parents and

To my brothers and sisters.

Zainab Ali Abu-Iram

Acknowledgement

I am very grateful to my supervisor Dr. Ali Zein for all his help and encouragement.

I am very grateful to my internal referee Dr. Mohammad Adm for his valuable suggestions on this thesis.

I thank my external referee Dr. Hasan Almanasreh for his useful comments and advice.

Also, my thanks to the members of the department of mathematics at the Palestine Polytechnic University.

Abstract

The Moore-Penrose inverse is one of the most important generalized inverses for arbitrary singular square or rectangular matrix. It finds many applications in engineering and applied sciences. The direct methods to find such inverse is expensive, especially for large matrices. Therefore, various numerical methods have been developed to compute the Moore-Penrose inverse.

This thesis is mainly concerned with the development of iterative methods to compute the Moore-Penrose inverse. Besides our new results the thesis contains several recent known iterative methods. The convergence properties of these methods are presented. And, several numerical examples are given.

Our own results involve new family of second-order iterative algorithms for computing the Moore-Penrose inverse. The construction of this algorithm is based on the usage of Penrose equations with approximations for p -th root for a product of the matrix with its inverse approximations. Convergence properties are considered. Numerical results are also presented and a comparison with Newton's method is made. It is observed that the new methods require less number of iterations than that of Newton's method. In addition, numerical experiments show that these methods are more effective than Newton's method when the number of columns increases than the number of rows.

In addition, we establish a new iterative scheme by using a square of the product of the matrix with its inverse approximations. By convergence analysis, we show that this scheme is also a second order. Several numerical tests are made. It is observed that the above family is more effective than this method.

Contents

1	Introduction	1
1.1	Overview	1
1.2	New results	4
1.3	Outline	4
2	Preliminaries	6
2.1	Singular value decomposition	6
2.2	Moore-Penrose inverse	11
2.3	Least squares solution	14
2.4	Balancing chemical equations	17
2.5	Basic definitions and lemmas	19
3	Iterative methods	21
3.1	First and second order iterative methods	21
3.2	Higher order iterative method	29
3.3	Fourth order iterative method	37
3.4	Other iterative methods	44
3.5	The choices of the initial value X_0	45
4	Our new iterative methods	47
4.1	A new family of second-order iterative methods	47
4.2	Second order iterative method	58

Chapter 1

Introduction

1.1 Overview

The inverse of a matrix appears frequently in various areas of applied mathematics and engineering systems. In many applications, the matrix appears as singular or rectangular. The theory of generalized inverse has attracted a considerable attention over the past few decades in order to deal with such matrices. The most important generalized inverses is the so called *Moore-Penrose inverse*.

Moore-Penrose inverse initially introduced in 1920 by Moore and independently rediscovered by Penrose in 1955 [1]. Since then, it has been extensively studied by many researchers and many methods are proposed to compute it.

The Moore-Penrose inverse of an $m \times n$ complex matrix A , denoted by A^\dagger , is a unique $n \times m$ matrix X satisfying the following four Penrose equations

$$(1) AXA = A, \quad (2) XAX = X, \quad (3) (AX)^* = AX, \quad (4) (XA)^* = XA,$$

where A^* the conjugate transpose of a matrix A .

The importance of the Moore-Penrose inverse arises in several applications, for example, in statistical regression analysis, prediction theory, control of robot manipulators as well as signal and image processing, see [3, 11, 18, 19]. Recently, Soleimani et.al. [24] used the Moore-Penrose inverse for balancing chemical equations.

In the last decades, many important methods to find the Moore-Penrose inverse of a matrix have been developed. Both direct and iterative methods can be used to compute the Moore-Penrose inverse.

One of the most commonly used direct methods is the *Singular Value Decomposition (SVD)* method, see e.g. [1, 7]. This method is very accurate but time intensive since it requires a large amount of computational resources, especially in the case of large matrices.

An alternative and very important approach to calculate the Moore-Penrose inverse is to use iterative methods. Iterative methods have attracted more attention in recent years, see [5, 9, 13, 26–28] and the references cited there in.

The most frequently used iterative methods for approximating the inverse of a matrix $A \in \mathbb{C}^{m \times n}$ is the famous *Newton's Method* [21]

$$X_{k+1} = X_k(2I - AX_k), \quad k = 0, 1, 2, \dots \quad (1.1)$$

This method is a second-order iterative methods. Shultz in [21] found that the eigenvalues of $I - AX_0$ must have magnitude less than 1 to ensure the convergence, where X_0 is the initial choice.

Li et al. in [9] investigated the following third-order method, known as *Chebyshev's Method*,

$$X_{k+1} = X_k(3I - AX_k(3I - AX_k)), \quad k = 0, 1, 2, \dots \quad (1.2)$$

Esmaeili et. al. in [5] proposed new fourth-order method to compute the Moore-Penrose inverse as follows

$$X_{k+1} = X_k[9I - 26(AX_k) + 34(AX_k)^2 - 21(AX_k)^3 + 5(AX_k)^4], \quad k = 0, 1, 2, \dots \quad (1.3)$$

Among many such matrix iterative methods, the hyper-power iteration of the order p is defined by the following scheme, see e.g. [26]

$$X_{k+1} = X_k(I + R_k + \cdots + R_k^{p-1}) = X_k \sum_{i=0}^{p-1} R_k^i, \quad R_k = I - AX_k. \quad (1.4)$$

The iterative Equation (1.4) requires p matrix-matrix multiplications to achieve the p -th order of convergence. The construction of this algorithm is based on Penrose equation (2).

It is clear that the class of higher order iterative methods [26] coincide with Newton's method if the order is reduced to 2. Moreover, if $p = 3$ it reduces to Chebyshev method.

In 2018, Pan et. al. [15] proposed a matrix iterative method based on (1.4) which achieve 18th order of convergence by using only seven matrix multiplications per iteration loop.

In addition, the authors of [5, 23, 24] were able to reduce the number of matrix multiplications for higher order methods using (1.4).

Moreover, various iterative methods have been developed based on the matrix equation $f(X) = X^{-1} - A = 0$, see e.g. [9, 23].

This thesis is mainly concerned with the iterative methods for computing the Moore-Penrose inverse. Initially, we present basic definition of the Moore-Penrose inverse and basic properties. Then, we consider the problem of balancing chemical equations as an applications for the Moore Penrose inverse. In addition, several known iterative methods are given. After that, we introduce our new iterative methods for computing the Moore Penrose inverse. Several numerical experiments are made to show the applicability of our results.

1.2 New results

Our attention is devoted to find a fast and efficient iterative method to compute the Moore-Penrose inverse. We use the idea of $A^{\frac{1}{2}}A^{\frac{1}{2}} = A$ and the Penrose equations (1) and (2) to propose the following iterative method

$$X_{k+1} = X_k - 2X_k((AX_k)^{\frac{1}{2}} - I), \quad (1.5)$$

after studying the convergence we find this method is a second-order iterative method. Moreover, we study the properties of this method.

In addition, by using the idea of (1.5) we construct the family of second-order iterative methods to compute the Moore-Penrose inverse as

$$X_{k+1} = X_k - pX_k((AX_k)^{\frac{1}{p}} - I), \quad p \in \{2, 3, 4, \dots\}. \quad (1.6)$$

These methods are written in terms of p -th root of a square matrix AX_k . Then approximations for the p -th root of a square matrix is used in computation. A wide set of numerical tests show that these methods require less number of iterations than Newton's method. In addition, numerical experiments show that these methods are more efficient than Newton's method when the number of columns increases than the number of rows. In this case the CPU time of our methods is also less than Newton's method.

In addition, we propose other iterative method written as

$$X_{k+1} = X_k - \beta X_k((AX_k)^2 - I). \quad (1.7)$$

We prove that this method converges to the Moore-Penrose inverse and its convergence is quadratic if $\beta = \frac{1}{2}$. Also, we discuss the properties of this method.

1.3 Outline

This thesis is organized as follows

Chapter 1 is an introduction chapter. It contains a short overview and a brief of our new results.

Then *Chapter 2* considers the basic definitions and basic properties of the Moore-Penrose inverse. In addition, we present the theory of the singular value decomposition for the computation of the Moore-Penrose inverse. Moreover, we give a practical application for the Moore-Penrose inverse in chemical balancing equations.

Chapter 3 is devoted to some known iterative methods for computing the Moore-Penrose inverse. Convergence properties of these methods are presented. Also, we give numerical examples.

In *Chapter 4*, our new iterative methods are introduced. Two approaches are established. Convergence analysis is studied and a set of numerical tests are given to investigate the new methods.

Chapter 2

Preliminaries

In this chapter, we present the singular value decomposition as a direct method to compute the Moore-Penrose inverse. Then, Penrose equations and some properties of the Moore-Penrose inverse are discussed. After that, least squares solutions are presented as an application for the Moore-Penrose inverse. Indeed, linear least squares problems occur in solving *overdetermined linear systems*, i.e. we are given more equations than unknowns. In general, such an overdetermined system has no solution, but we may find a meaningful approximate solution by minimizing some norm of the residual vector. This solution is computed by using the Moore-Penrose inverse.

In addition, we present a practical application for the Moore-Penrose inverse in balancing chemical equations. At the end of this chapter, some basic definitions and lemmas which are essential in the rest of our thesis are given.

2.1 Singular value decomposition

In this section, we present a computationally simple and accurate way to compute the Moore-Penrose inverse of a matrix by using the singular value decomposition (SVD), see [1, 7].

For each $m \times n$ matrix A , the $n \times n$ matrix A^*A is Hermitian and positive semidefinite. Therefore, the eigenvalues of A^*A are real and nonnegative.

For an $m \times n$ matrix A of rank r , the null space $\mathcal{N}(A) = \{x \in \mathbb{C}^n \mid Ax = 0\}$ has dimension $\dim(\mathcal{N}(A)) = n - r$.

Definition 2.1.1. *The nonnegative square roots of the eigenvalues of A^*A are called the singular values of A .*

Lemma 2.1.1. *([7]) For an $n \times n$ matrix A and its adjoint A^* we have*

$$(Ax, y) = (x, A^*y),$$

for all $x, y \in \mathbb{C}^n$, where (\cdot, \cdot) denotes the Euclidean scalar product, $(x, y) = \sum_{j=1}^n x_j \bar{y}_j$.

Theorem 2.1.2. *([7]) Let A be an $m \times n$ matrix of rank r . Then there exist nonnegative numbers*

$$\mu_1 \geq \mu_2 \geq \dots \geq \mu_r > \mu_{r+1} = \dots = \mu_n = 0,$$

and orthonormal vectors $u_1, \dots, u_n \in \mathbb{C}^n$ and $v_1, \dots, v_m \in \mathbb{C}^m$ such that

$$Au_j = \mu_j v_j, \quad A^*v_j = \mu_j u_j, \quad j = 1, \dots, r, \quad (2.1a)$$

$$Au_j = 0, \quad j = r + 1, \dots, n, \quad (2.1b)$$

$$A^*v_j = 0, \quad j = r + 1, \dots, m. \quad (2.1c)$$

For each $x \in \mathbb{C}^n$ we have the singular value decomposition

$$Ax = \sum_{j=1}^r \mu_j (x, u_j) v_j. \quad (2.2)$$

Each system (μ_j, u_j, v_j) with these properties is called a singular system of the matrix A .

Proof. The Hermitian and semipositive definite matrix A^*A of rank r has n orthonormal eigenvectors u_1, \dots, u_n with nonnegative eigenvalues μ_1^2, \dots, μ_n^2 , i.e.

$$A^*Au_j = \mu_j^2 u_j, \quad j = 1, \dots, n, \quad (2.3)$$

where μ_j can be ordered as

$$\mu_1 \geq \mu_2 \geq \dots \geq \mu_r > 0 \text{ and } \mu_{r+1} = \dots = \mu_n = 0.$$

Define

$$v_j := \frac{1}{\mu_j} Au_j, \quad j = 1, \dots, r.$$

Then,

$$\begin{aligned}(v_j, v_k) &= \left(\frac{1}{\mu_j} Au_j, \frac{1}{\mu_k} Au_k \right) = \frac{1}{\mu_j \mu_k} (u_j, A^* Au_k) \\ &= \frac{1}{\mu_j \mu_k} (u_j, \mu_k^2 u_k) \\ &= \frac{\mu_k}{\mu_j} (u_j, u_k) \\ &= \begin{cases} 1 & j = k, \\ 0 & j \neq k. \end{cases}\end{aligned}$$

Hence, v_1, \dots, v_m are orthonormal.

Now, for (2.1a)

$$\begin{aligned}A^* v_j &= A^* \frac{1}{\mu_j} Au_j \\ &= \frac{1}{\mu_j} A^* Au_j \\ &= \frac{1}{\mu_j} \mu_j^2 u_j = \mu_j u_j, \quad j = 1, \dots, r.\end{aligned}$$

And from $v_j = \frac{1}{\mu_j} Au_j$, we have

$$Au_j = \mu_j v_j, \quad j = 1, \dots, r.$$

Hence (2.1a) is proved.

Since $\mathcal{N}(A) = \mathcal{N}(A^*A)$, then for $j = r + 1, \dots, m$ we have

$$Au_j = 0.$$

If $r < m$, by Gram-Schmidt orthogonalization procedure we can extend v_1, \dots, v_r to an orthonormal basis v_1, \dots, v_m of \mathbb{C}^m .

Since A^* has rank r , we have

$$\dim(\mathcal{N}(A^*)) = m - r.$$

From this we can conclude (2.1c).

Since u_1, \dots, u_n form an orthonormal basis of \mathbb{C}^n , then for $x \in \mathbb{C}^n$

$$x = \sum_{j=1}^n (x, u_j) u_j,$$

then

$$\begin{aligned} Ax &= \sum_{j=1}^n (x, u_j) Au_j \\ &= \sum_{j=1}^r (x, u_j) \mu_j v_j \\ &= \sum_{j=1}^r \mu_j (x, u_j) v_j. \end{aligned}$$

□

Clearly, we can rewrite the Equations (2.1) in the form

$$A = VDU^*,$$

where $U = (u_1, \dots, u_n)$ and $V = (v_1, \dots, v_m)$ are unitary $n \times n$ and $m \times m$ matrices, respectively, and where D is an $m \times n$ diagonal matrix with entries

$$d_{jj} = \begin{cases} \mu_j & j = 1, \dots, r, \\ 0 & \text{otherwise.} \end{cases}$$

Example 2.1. Consider the matrix $A = \begin{bmatrix} 0 & 1 & 1 \\ \sqrt{2} & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$.

To compute its SVD, firstly, we compute the eigenvalues of A^*A .

$$A^*A = \begin{bmatrix} 2 & 2\sqrt{2} & 0 \\ 2\sqrt{2} & 6 & 2 \\ 0 & 2 & 2 \end{bmatrix}$$

The matrix A^*A has eigenvalues $\mu_1^2 = 8$, $\mu_2^2 = 2$ and $\mu_3^2 = 0$ and eigenvectors

$$u_1 = \begin{bmatrix} \frac{1}{\sqrt{6}} \\ \frac{3}{\sqrt{12}} \\ \frac{1}{\sqrt{12}} \end{bmatrix}, \quad u_2 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ 0 \\ -\frac{2}{\sqrt{6}} \end{bmatrix} \text{ and } u_3 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}.$$

Now, we find v_1 , v_2 and v_3 by using $Au_j = \mu_j v_j$ to get

$$v_1 = \begin{bmatrix} \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \end{bmatrix}, \quad v_2 = \begin{bmatrix} -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \end{bmatrix} \text{ and } v_3 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Hence, we have

$$U = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \\ \frac{3}{\sqrt{12}} & 0 & -\frac{1}{2} \\ \frac{1}{\sqrt{12}} & -\frac{2}{\sqrt{6}} & \frac{1}{2} \end{bmatrix}, \quad V = \begin{bmatrix} \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \text{ and } D = \begin{bmatrix} 2\sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then,

$$A = VDU^* = \begin{bmatrix} \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 2\sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{3}{\sqrt{12}} & \frac{1}{\sqrt{12}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

2.2 Moore-Penrose inverse

The Moore-Penrose inverse is a generalization of the inverse of a non-singular matrix, we present the Moore-Penrose inverse by using singular value decomposition. Moreover, we introduce the Penrose equations. We conclude this section with basic properties of the Moore-Penrose inverse.

Definition 2.2.1. Let $A = VDU^*$ be the singular value decomposition with

$$D = \begin{pmatrix} D_r & \\ & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad D_r := \text{diag}(\mu_1, \dots, \mu_r, 0, \dots, 0) \in \mathbb{R}^{n \times n}$$

with $\mu_1 \geq \dots \geq \mu_r > 0$. Then the matrix $A^\dagger = UD^\dagger V^*$ with

$$D^\dagger = \begin{pmatrix} D_r^\dagger & 0 \\ & \end{pmatrix} \in \mathbb{R}^{n \times m}, \quad D_r^\dagger := \text{diag}\left(\frac{1}{\mu_1}, \dots, \frac{1}{\mu_r}, 0, \dots, 0\right) \in \mathbb{R}^{n \times n}$$

is called the Moore-Penrose inverse of A .

Theorem 2.2.2. (Penrose Equations) $X = A^\dagger$ is the only solution of the matrix equations

$$AXA = A, \tag{2.4a}$$

$$XAX = X, \tag{2.4b}$$

$$(AX)^* = AX, \tag{2.4c}$$

$$(XA)^* = XA. \tag{2.4d}$$

Proof. To verify that A^\dagger is a solution, inserting the SVD to get the result.

For (2.4a),

$$AA^\dagger A = VDU^*UD^\dagger V^*VDU^* = VDD^\dagger DU^* = VDU^* = A.$$

We can complete the proof of (2.4b), (2.4c) and (2.4d) by the same idea. To prove uniqueness, assume that X is any solution to (2.4). Then

$$\begin{aligned} X &= XAX \\ &= (XA)^*X = A^*X^*X \\ &= (AA^\dagger A)^*X^*X = A^*(A^\dagger)^*A^*X^*X \\ &= A^*(A^\dagger)^*XAX \\ &= A^*(A^\dagger)^*X = (A^\dagger A)^*X \\ &= A^\dagger AX \\ &= A^\dagger AA^\dagger AX \\ &= A^\dagger(AA^\dagger)^*(AX)^* = A^\dagger(A^\dagger)^*A^*X^*A^* \\ &= A^\dagger(A^\dagger)^*A^* \\ &= A^\dagger(AA^\dagger)^* = A^\dagger AA^\dagger \\ X &= A^\dagger \end{aligned}$$

□

Equations (2.4) are the well-known Moore-Penrose equations.

Proposition 2.1. *The Moore-Penrose inverse satisfies the following relations*

$$A^\dagger = A^\dagger(A^\dagger)^*A^*, \tag{2.5a}$$

$$A = AA^*(A^\dagger)^*, \tag{2.5b}$$

$$A^* = A^*AA^\dagger, \tag{2.5c}$$

$$A^\dagger = A^*(A^\dagger)^*A^\dagger, \quad (2.5d)$$

$$A = (A^\dagger)^*A^*A, \quad (2.5e)$$

$$A^* = A^\dagger AA^*, \quad (2.5f)$$

for all $A \in \mathbb{C}^{m \times n}$.

Proof. Since $AA^\dagger = (AA^\dagger)^*$, multiplying from left by A^\dagger , we get

$$A^\dagger = A^\dagger(A^\dagger)^*A^*, \text{ which is Equation (2.5a).}$$

Replacing $A \rightarrow A^\dagger$ and using the fact that $A = (A^\dagger)^\dagger$, then Equation (2.5a), leads to

$$A = AA^*(A^\dagger)^*, \text{ which is (2.5b).}$$

Replacing $A \rightarrow A^*$ and using the fact that $(A^*)^\dagger = (A^\dagger)^*$, then Equation (2.5b), gives

$$A^* = A^*AA^\dagger, \text{ which is (2.5c).}$$

Relations (2.5d)-(2.5f) can be obtained from the fact that $A^\dagger A = (A^\dagger A)^*$, and we can complete the proof by the same way. \square

Example 2.2. Let us consider the matrix $A = \begin{bmatrix} 0 & 1 & 1 \\ \sqrt{2} & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ that appeared in Example 2.1.

The Moore-Penrose inverse is

$$\begin{aligned} A^\dagger &= UD^\dagger V^* \\ &= \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \\ \frac{3}{\sqrt{12}} & 0 & -\frac{1}{2} \\ \frac{1}{\sqrt{12}} & -\frac{2}{\sqrt{6}} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} -\frac{1}{4\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{4\sqrt{2}} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{9}{24} & -\frac{1}{4} & \frac{9}{24} \end{bmatrix}.$$

2.3 Least squares solution

In this section, we consider one of the main applications of the Moore-Penrose inverse, namely the optimization of linear least square problems. In many situations, a solution of a linear system is non-existing or non-unique, but we want to find a vector x such that the norm of the difference $Ax - y$ is the smallest possible. For the material of this section we refer to [1, 6, 7].

Theorem 2.3.1. ([7]) *Let A be an $m \times n$ matrix of rank r with singular system (μ_j, u_j, v_j) . The linear system*

$$Ax = y, \tag{2.6}$$

is solvable if and only if

$$(y, z) = 0,$$

*for all $z \in \mathbb{C}^m$ with $A^*z = 0$. In this case a solution of (2.6) is given by*

$$x_0 = \sum_{j=1}^r \frac{1}{\mu_j} (y, v_j) u_j. \tag{2.7}$$

Proof. Let x be a solution of (2.6) and let $A^*z = 0$, then

$$(y, z) = (Ax, z) = (x, A^*z) = (x, 0) = 0.$$

Conversely, assume the condition $(y, z) = 0$ for all $z \in \mathbb{C}^m$ with $A^*z = 0$ is satisfied. Since v_1, \dots, v_m is a basis for \mathbb{C}^m , then

$$y = \sum_{j=1}^m (y, v_j) v_j$$

$$= \sum_{j=1}^r (y, v_j) v_j.$$

Now,

$$\begin{aligned} Ax_0 &= \sum_{j=1}^r \frac{1}{\mu_j} (y, v_j) Au_j \\ &= \sum_{j=1}^r \frac{1}{\mu_j} (y, v_j) \mu_j v_j \\ &= \sum_{j=1}^r (y, v_j) v_j \\ &= y. \end{aligned}$$

Hence, x_0 is a solution of (2.6). □

Since $\mathcal{N}(A) = \text{span}\{u_{r+1}, \dots, u_n\}$ the vector x_0 defined by (2.7) has the property

$$(x_0, x) = 0,$$

for all $x \in \mathcal{N}(A)$.

In the case where Equation (2.6) has more than one solution, the general solution is obtained from (2.7) by adding an arbitrary solution x of the homogeneous equation $Ax = 0$. Then

$$\begin{aligned} \|x_0 + x\|_2^2 &= (x_0 + x, x_0 + x) \\ &= \|x_0\|_2^2 + (x, x_0) + (x_0, x) + \|x\|_2^2 \\ &= \|x_0\|_2^2 + \|x\|_2^2. \end{aligned}$$

Hence,

$$\|x_0\|_2^2 \leq \|x_0 + x\|_2^2$$

We observe that (2.7) represents the uniquely determined solution of (2.6) with minimal Euclidean norm.

In the case where Equation (2.6) is not solvable, let

$$y = \sum_{j=1}^m (y, v_j) v_j,$$

Let x_0 be given by (2.7) and let $x \in \mathbb{C}^n$ be arbitrary. Then

$$(Ax - Ax_0, Ax_0 - y) = 0,$$

since $Ax - Ax_0 \in \text{span}\{v_1, \dots, v_r\}$ and $Ax_0 - y \in \text{span}\{v_{r+1}, \dots, v_m\}$.

This implies

$$\begin{aligned} \|Ax - y\|_2^2 &= (Ax - y, Ax - y) \\ &= (Ax - Ax_0 + Ax_0 - y, Ax - Ax_0 + Ax_0 - y) \\ &= \|Ax - Ax_0\|_2^2 + \|Ax_0 - y\|_2^2. \end{aligned}$$

Hence,

$$\|Ax_0 - y\|_2^2 \leq \|Ax - y\|_2^2.$$

Again, it can be shown that x_0 is the uniquely determined least squares solution with minimal Euclidean norm.

Hence, (2.7) defines a linear operator $A^\dagger : \mathbb{C}^m \rightarrow \mathbb{C}^n$ by

$$A^\dagger y := \sum_{j=1}^r \frac{1}{\mu_j} (y, v_j) u_j, \quad y \in \mathbb{C}^m.$$

Theorem 2.3.2. ([6]) *The general solution of the linear least squares problem $Ax \approx y$ is*

$$x = A^\dagger y + (I - A^\dagger A)w, \quad w \in \mathbb{C}^n \text{ arbitrary.} \quad (2.8)$$

If we calculate $\|x\|_2^2$ using (2.8) and by using (2.5d), we obtain

$$\begin{aligned} \|x\|_2^2 &= \|A^\dagger y\|_2^2 + (A^\dagger y, (I - A^\dagger A)w) + ((I - A^\dagger A)w, A^\dagger y) + \|(I - A^\dagger A)w\|_2^2 \\ &= \|A^\dagger y\|_2^2 + w^T (I - A^\dagger A)^T A^\dagger y + y^T A^{\dagger T} (I - A^\dagger A)w + \|(I - A^\dagger A)w\|_2^2 \\ &= \|A^\dagger y\|_2^2 + w^T (A^\dagger - A^T A^{\dagger T} A^\dagger) y + y^T (A^{\dagger T} - A^{\dagger T} A^\dagger A) w + \|(I - A^\dagger A)w\|_2^2 \end{aligned}$$

$$\begin{aligned}
&= \|A^\dagger y\|_2^2 + \|(I - A^\dagger A)w\|_2^2 \\
&\geq \|A^\dagger y\|_2^2.
\end{aligned}$$

Any solution to the least squares problem must have norm greater than or equal to $\|A^\dagger y\|_2$, this means that, the Moore-Penrose inverse produces the minimum norm solution to the least squares problem $Ax \approx y$.

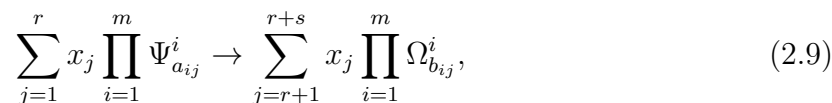
If A has full rank, then the solution of the linear least square problem is unique

$$x = A^\dagger y = A^{-1}y.$$

2.4 Balancing chemical equations

This section is devoted to the applicability of Moore-Penrose inverse in balancing chemical equations. The main idea is due to Soleimani et.al. [24].

It is assumed that a chemical system is modeled by a single reaction of the general form, see e.g. [17]



where x_j , $j = 1, \dots, r$ (x_j , $j = r + 1, \dots, r + s$) are unknown rational coefficients of the reactants (the products), Ψ^i , Ω^i , $i = 1, \dots, m$ are chemical elements in reactants and products, respectively,

and, a_{ij} , $i = 1, \dots, m$, $j = 1, \dots, r$ and b_{ij} , $i = 1, \dots, m$, $j = r + 1, \dots, r + s$ are the numbers of atoms Ψ^i and Ω^i , respectively, in the j -th molecule.

It is necessary to form an $m \times n$ matrix A , called the *reaction matrix*, whose columns represent the reactants and products and the rows represent the distinct atoms in the chemical reaction. More precisely, the (i, j) -th element of A represents the number of atoms of type i in each element (reactant or product). An arbitrary element a_{ij} is positive or negative according to whether it corresponds to a reactant or a product.

Hence, the balancing chemical equation problem can be formulated as the homogeneous matrix equation

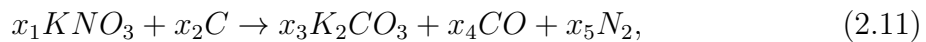
$$Ax = 0.$$

The general solution of the balancing problem is given by

$$x = (I - A^\dagger A)w, \quad w \in \mathbb{C}^n \text{ arbitrary.} \quad (2.10)$$

Hence, we can find the exact solution of balancing chemical equation by finding the Moore-Penrose inverse A^\dagger . Then, compute the vector x using Equation (2.10). Finally, transform real numbers included in x into an exact solution by divide x by the minimum of numerators in x .

Example 2.3. Consider a specific skeletal chemical equation from [20]



where the left hand side of the arrow consists of elements called reactants, while the right hand side comprises elements called the products.

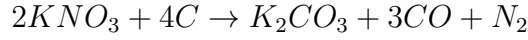
Hence, (2.11) is formulated as the homogeneous equation $Ax = 0$, wherein 0 denotes the null column vector and

$$A = \begin{bmatrix} 1 & 0 & -2 & 0 & 0 \\ 1 & 0 & 0 & 0 & -2 \\ 3 & 0 & -3 & -1 & 0 \\ 0 & 1 & -1 & -1 & 0 \end{bmatrix}$$

By using `pinv()` command in MATLAB, we get

$$A^\dagger = \begin{bmatrix} -0.5161 & 0.0323 & 0.4516 & -0.2581 \\ -0.0323 & 0.0645 & -0.0968 & 0.4839 \\ -0.7581 & 0.0161 & 0.2258 & -0.1290 \\ 0.7258 & 0.0484 & -0.3226 & -0.3871 \\ -0.2581 & -0.4839 & 0.2258 & -0.1290 \end{bmatrix}$$

The final exact coefficients are defined as $(x_1, x_2, x_3, x_4, x_5)^T = (2, 4, 1, 3, 1)^T$. Thus,



2.5 Basic definitions and lemmas

This section contains basic definitions and lemmas that are needed later in the thesis.

Definition 2.5.1. ([2]) A sequence $\{X_k\}_{k=1}^{\infty}$ of matrices in $\mathbb{C}^{m \times n}$ is said to converge to a matrix X with respect to the norm $\|\cdot\|$ if, given any $\epsilon > 0$, there exists an integer $N(\epsilon)$ such that

$$\|X_k - X\| < \epsilon, \quad \text{for all } k \geq N(\epsilon).$$

Definition 2.5.2. ([2]) Suppose $\{X_k\}_{k=1}^{\infty}$ is a sequence of matrices that converges to X , with $X_k \neq X$ for all k . If positive constants λ and α exist with

$$\lim_{k \rightarrow \infty} \frac{\|X_{k+1} - X\|}{\|X_k - X\|^\alpha} = \lambda,$$

then $\{X_k\}_{k=1}^{\infty}$ converges to X of order α , with asymptotic error constant λ .

Definition 2.5.3. ([2]) The $n \times n$ matrix A is said to be diagonally dominant when

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}| \quad \text{holds for each } i = 1, 2, \dots, n.$$

Definition 2.5.4. ([2]) A matrix $X \in \mathbb{C}^{n \times n}$ is a p -th root of a matrix $A \in \mathbb{C}^{n \times n}$ if and only if $X^p = A$.

Definition 2.5.5. ([2]) The number x is a fixed point for a given function g if $g(x) = x$.

Definition 2.5.6. ([2]) If $A = (a_{ij})$ is an $n \times n$ matrix, then $\|A\|_F = \sqrt{\text{trace}(AA^*)}$.

Lemma 2.5.1. ([7]) Let $M \in \mathbb{C}^{n \times n}$ and $\epsilon > 0$ be given. There is at least one matrix norm $\|\cdot\|$ such that

$$\rho(M) \leq \|M\| \leq \rho(M) + \epsilon,$$

where $\rho(M) = \max\{|\lambda_1(M)|, \dots, |\lambda_n(M)|\}$ denotes the spectral radius of M .

Lemma 2.5.2. ([27]) If $P \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{n \times n}$ are such that $P = P^2$ and $PQ = QP$, then

$$\rho(PQ) \leq \rho(Q).$$

Lemma 2.5.3. ([2]) Let $g \in C[a, b]$ be such that $g(x) \in [a, b]$, for all x in $[a, b]$. Suppose, in addition, that g' exists on (a, b) and that a constant $0 < k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a, b).$$

Then for any number p_0 in $[a, b]$, the sequence defined by

$$p_n = g(p_{n-1}), \quad n \geq 1,$$

converges to the unique fixed point p in $[a, b]$.

Lemma 2.5.4. ([2]) Let p be a solution of the equation $x = g(x)$. Suppose that $g'(p) = 0$ and g'' is continuous with $|g''| < M$ on an open interval I containing p . Then there exists a $\delta > 0$ such that, for $p_0 \in [p - \delta, p + \delta]$, the sequence defined by $p_n = g(p_{n-1})$, $n \geq 1$, converges at least quadratically to p . Moreover, for sufficiently large values of n ,

$$|p_{n+1} - p_n| < \frac{M}{2} |p_n - p|^2.$$

Chapter 3

Iterative methods

In this chapter, we consider some known iterative methods for computing the Moore-Penrose inverse. We start by presenting first and second-order iterative methods developed by Petkovic and Stanimirovic [12, 13]. The construction of this algorithm is based on Penrose Equations (2.4b) and (2.4d). Then, we present a class of higher order iterative method [26] based on Penrose Equation (2.4b) and by extending the iterative method in [13]. Next, we present different iterative methods, from [8, 24, 25, 28], to compute the Moore-Penrose inverse. Convergence properties of these algorithm are also considered.

3.1 First and second order iterative methods

Petkovic and Stanimirovic in [13] presented first and second-order iterative methods for computing Moore-Penrose inverse. In [12] the authors corrected and improved the method in [13]. We consider the properties of this algorithm. Numerical examples are also presented.

Let $A \in \mathbb{C}^{m \times n}$ and $X = A^\dagger \in \mathbb{C}^{n \times m}$. We use Equations (2.4b) and (2.4d) and obtain

$$X^* = (XAX)^* = X^*(XA)^* = X^*XA.$$

Hence, for arbitrary $\beta \in \mathbb{R}$ it holds

$$X^* = X^* - \beta(X^*XA - X^*) = X^*(I - \beta XA) + \beta X^*,$$

or equivalently

$$X = (I - \beta XA)^* X + \beta X.$$

From the last equation we get the following iterative method

$$X_{k+1} = (I - \beta X_k A)^* X_k + \beta X_k. \quad (3.1)$$

Assume the starting value of the iterative method (3.1) is

$$X_0 = \alpha A^*, \quad (3.2)$$

for an appropriate real number α .

The following lemma will be used for establishing the convergence of the iterative method.

Lemma 3.1.1. *For the sequence X_k generated by the iterative schemes (3.1) and (3.2) the following holds*

$$(X_k A)^* = (X_k A), \quad (3.3a)$$

$$X_k A X = X_k, \quad (3.3b)$$

$$X A X_k = X_k, \quad (3.3c)$$

where $k \geq 0$.

Proof. We use mathematical induction. For $k = 0$ we have $X_0 = \alpha A^*$ and all statements in (3.3) hold. Assume the statements are true for some integer k . Now we prove the statements for $k + 1$.

For (3.3a), we have

$$\begin{aligned} (X_{k+1} A)^* &= ((I - \beta X_k A)^* X_k A + \beta X_k A)^* \\ &= (X_k A)^* (I - \beta X_k A) + \beta (X_k A)^* \\ &= X_k A (I - \beta X_k A) + \beta X_k A \end{aligned}$$

$$\begin{aligned} &= X_k A - \beta X_k A X_k A + \beta X_k A \\ &= (I - \beta X_k A) X_k A + \beta X_k A \\ &= (I - \beta X_k A)^* X_k A + \beta X_k A \\ &= X_{k+1} A. \end{aligned}$$

We prove (3.3b) in a similar way

$$\begin{aligned} X_{k+1} A X &= ((I - \beta X_k A)^* X_k + \beta X_k) A X \\ &= (I - \beta X_k A)^* X_k A X + \beta X_k A X \\ &= (I - \beta X_k A)^* X_k + \beta X_k \\ &= X_{k+1}. \end{aligned}$$

Now, for (3.3c)

$$\begin{aligned} X A X_{k+1} &= X A (I - \beta X_k A)^* X_k + \beta X A X_k \\ &= X A (I - \beta X_k A) X_k + \beta X_k \\ &= X A X_k - \beta X A X_k A X_k + \beta X_k \\ &= X_k - \beta X_k A X_k + \beta X_k \\ &= (I - \beta X_k A) X_k + \beta X_k \\ &= (I - \beta X_k A)^* X_k + \beta X_k \\ &= X_{k+1}. \end{aligned}$$

This completes the proof of the lemma. □

From Lemma 3.1.1, the scheme (3.1) can be written as

$$\begin{aligned} X_{k+1} &= (1 - \beta X_k A)^* X_k + \beta X_k \\ &= (1 - \beta X_k A) X_k + \beta X_k \end{aligned}$$

$$= (1 + \beta)X_k - \beta X_k A X_k. \quad (3.4)$$

Now, we want to prove that the matrix sequence X_k defined by the iterative method (3.4) and the starting value (3.2), converges to the Moore-Penrose inverse $X = A^\dagger$.

Theorem 3.1.1. *Iterative method (3.4) with the starting value defined in (3.2) converges to the Moore-Penrose inverse $X = A^\dagger$ under the assumptions*

$$\|(X_0 - X)A\| < 1, \quad 0 < \beta \leq 1. \quad (3.5)$$

For $\beta < 1$ the method has a linear convergence, while for $\beta = 1$ its convergence is quadratic.

Proof. Using Lemma 3.1.1 and substituting for X_k , we get

$$\|E_k\| = \|X_k - X\| = \|X_k A X - X A X\| \leq \|X_k A - X A\| \|X\| = \|t_k\| \|X\|,$$

where $t_k = X_k A - X A = E_k A$.

Now, by using Lemma 3.1.1 with (3.4), we get

$$\begin{aligned} t_{k+1} &= X_{k+1} A - X A = (1 + \beta)X_k A - \beta X_k A X_k A - X A \\ &= X_k A + \beta X_k A - \beta (X_k A)^2 - (X A)^2 \\ &= -[-X A X_k A - \beta X_k A X A + \beta (X_k A)^2 + (X A)^2] \\ &= -(\beta X_k A - X A)(X_k A - X A) \\ &= -(\beta(X_k A - X A) - (1 - \beta)X A)(X_k A - X A) \\ &= -\beta(X_k A - X A)^2 + (1 - \beta)(X_k A - X A) \\ &= -\beta t_k^2 + (1 - \beta)t_k. \end{aligned} \quad (3.6)$$

Let $s_k = \|t_k\|$, we require that $s_k \rightarrow 0$ as $k \rightarrow \infty$. Condition (3.5) implies $s_0 < 1$, then if we assume that $s_k < 1$, from (3.6) and inductive method, we obtain

$$s_{k+1} \leq \beta s_k^2 + (1 - \beta)s_k < \beta s_k + (1 - \beta)s_k < s_k < 1. \quad (3.7)$$

Thus, s_k is a monotonically decreasing bounded sequence converging to s as $k \rightarrow \infty$ and $0 \leq s < 1$. From (3.7), we get

$$s \leq \beta s^2 + (1 - \beta)s,$$

then

$$\beta s^2 - \beta s \geq 0.$$

This gives either $s = 0$ or $s \geq 1$. Thus, $s = 0$.

This completes the proof that $s_k \rightarrow 0$ as $k \rightarrow \infty$. Thus, $X_k \rightarrow X$ as $k \rightarrow \infty$.

From (3.6), we conclude that iterative method (3.4) has linear convergence if $\beta < 1$, and its convergence is quadratic if $\beta = 1$. \square

For Theorem 3.1.1 we need to write the condition (3.5) in an equivalent form which does not contain the Moore-Penrose inverse X .

According to Lemma 2.5.1, necessary and sufficient condition for the convergence of the iterative method is that $\rho((\alpha A^* - X)A) < 1$.

Theorem 3.1.2. *Let the eigenvalues of a matrix A^*A satisfy*

$$\sigma_1(A) \geq \dots \geq \sigma_r(A) > \sigma_{r+1}(A) = \dots = 0.$$

Condition $\rho((\alpha A^ - X)A) < 1$ is satisfied under the assumptions*

$$\max_{1 \leq i \leq r} |1 - \alpha \lambda_i(A^*A)| < 1,$$

where σ_i are the singular values of A .

Proof. Let $P = XA$ and $Q = \alpha A^*A - I$. Since $P^2 = P$ and

$$\begin{aligned} PQ &= \alpha XAA^*A - XA = \alpha(XA)^*A^*A - XA \\ &= \alpha A^*A - XA \\ &= \alpha A^*AXA - XA \end{aligned}$$

$$\begin{aligned}
 &= (\alpha A^* A - I) X A \\
 &= Q P,
 \end{aligned}$$

from Lemma 2.5.2 we can conclude that

$$\begin{aligned}
 \rho(PQ) &= \rho(XA(\alpha A^* A - I)) \\
 &= \rho((\alpha A^* - X)A) \\
 &\leq \rho(\alpha A^* A - I) \\
 &= \max_{1 \leq i \leq r} |1 - \alpha \lambda_i(A^* A)| < 1.
 \end{aligned}$$

□

Remark 3.1. Note that for $\beta = 1$ the method (3.4) reduces to the well-known Newton-Shultz method [21] for computing the Moore-Penrose inverse of a given matrix.

Lemma 3.1.2. Let $A \in \mathbb{C}^{m \times n}$. Sequence X_k defined by (3.4) and (3.2) satisfies $\mathcal{R}(X_k) = \mathcal{R}(A^*)$ and $\mathcal{N}(X_k) = \mathcal{N}(A^*)$, for each $k \geq 0$, where \mathcal{N}, \mathcal{R} are the null space and rank of matrix, respectively.

Proof. For $k = 0$, the statements of the lemma holds since $X_0 = \alpha A^*$.

Let $y \in \mathcal{N}(X_k)$ be an arbitrary vector. From (3.4) we have

$$X_{k+1}y = (1 + \beta)X_k y - \beta X_k A X_k y = 0.$$

Hence, $y \in \mathcal{N}(X_{k+1})$, which implies $\mathcal{N}(X_k) \subseteq \mathcal{N}(X_{k+1})$. Then, $\mathcal{R}(X_k) \geq \mathcal{R}(X_{k+1})$.

Hence, by mathematical induction we obtain

$$\mathcal{N}(X_k) \supseteq \mathcal{N}(X_0) = \mathcal{N}(A^*),$$

and

$$\mathcal{R}(X_k) \leq \mathcal{R}(X_0) = \mathcal{R}(A^*).$$

Now, to prove equality, let us consider $\mathcal{N} = \bigcup_{k \in \mathbb{N}_0} \mathcal{N}(X_k)$.

Let $y \in \mathcal{N}$ be an arbitrary vector and let $y \in \mathcal{N}(X_{k_0})$, for some $k_0 \in \mathbb{N}_0$.

Hence, $y \in \mathcal{N}(X_k)$ for all $k \geq k_0$, we have $X_k y = 0$ and using Theorem 3.1.1 we have

$$Xy = \lim_{k \rightarrow \infty} X_k y = 0.$$

This implies that $y \in \mathcal{N}(X) = \mathcal{N}(A^*)$ and we get $\mathcal{N} \subseteq \mathcal{N}(A^*)$.

Moreover, we have

$$\mathcal{N}(A^*) \subseteq \mathcal{N}(X_k) \subseteq \mathcal{N} \subseteq \mathcal{N}(A^*),$$

hence, we conclude that $\mathcal{N}(X_k) = \mathcal{N}(A^*)$.

Now,

$$\mathcal{R}(X_k) = m - \dim(\mathcal{N}(X_k)) = m - \dim(\mathcal{N}(A^*)) = \mathcal{R}(A^*).$$

Hence,

$$\mathcal{R}(X_k) = \mathcal{R}(A^*).$$

□

Next, numerical tests are given with a tolerance $\epsilon = 10^{-8}$, we use the termination criterion as in [26]

$$\text{res}(X) = \max\{\|AX_k A - A\|_F, \|X_k A X_k - X_k\|_F, \|(AX_k)^* - AX_k\|_F, \|(X_k A)^* - X_k A\|_F\} \leq \epsilon, \quad (3.8)$$

where, $\|\cdot\|_F$ the Frobenius norm of a matrix.

Example 3.1. Consider the matrix A of order (5×4) given by

$$A = \begin{bmatrix} 0.2794 & 0.1676 & 0.0645 & 0.2326 \\ 0.0065 & 0.2365 & 0.2274 & 0.1261 \\ 0.2271 & 0.1430 & 0.1009 & 0.2867 \\ 0.1265 & 0.1015 & 0.1806 & 0.2846 \\ 0.2773 & 0.0632 & 0.0503 & 0.1979 \end{bmatrix}.$$

The choice $\alpha = 0.6$ satisfies the convergence criterion given by

$$\max_{1 \leq i \leq 4} |1 - \alpha \lambda_i(A^* A)| = 0.9988 < 1,$$

since the eigenvalues of the matrix A^*A are

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0.0020, 0.0146, 0.0832, 0.6152).$$

The iterative method (3.4) with $\beta = 0.5$ generates a sequence of iterates $\{X_k\}$ after 42 steps converging to the Moore-Penrose inverse A^\dagger given by

$$A^\dagger = \begin{bmatrix} -0.2165 & 1.4802 & -4.9702 & -1.3732 & 8.4865 \\ 5.0277 & 1.8673 & 4.1653 & -4.6975 & -6.3778 \\ -5.3215 & 4.5524 & -8.4278 & 3.4688 & 10.5748 \\ 0.8566 & -4.0180 & 6.9330 & 3.0649 & -7.8449 \end{bmatrix}.$$

If we use $\beta = 0.3$ the method (3.4) needs 73 iterations and with $\beta = 0.8$ it needs 24 iterations. While Newton method needs 14 iterations to have the same result.

Example 3.2. Consider the ill-conditional Hilbert matrix A of order (5×5) given by

$$A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

The choice $\alpha = 0.8$ satisfies the convergence criterion given by

$$\max_{1 \leq i \leq 3} |1 - \alpha \lambda_i(A^*A)| = 0.9999 < 1,$$

since the eigenvalues of the matrix A^*A are

$$(\lambda_1, \lambda_2, \lambda_3) = (2.4556, 0.0435, 0.0001).$$

The iterative method (3.4) generates a sequence of iterates $\{X_k\}$ converging to the Moore-Penrose inverse A^\dagger given by

$$A^\dagger = \begin{bmatrix} 25 & -300 & 1050 & -1400 & 630 \\ -300 & 4800 & -18900 & 26880 & -12600 \\ 1050 & -18900 & 79380 & -117600 & 56700 \\ -1400 & 26880 & -117600 & 179200 & -88200 \\ 630 & -12600 & 56700 & -88200 & 44100 \end{bmatrix}.$$

When we use $\beta = 0.8$ the iterative method (3.4) needs 53 iteration, but when $\beta = 1$ it needs 42 iteration.

3.2 Higher order iterative method

We now describe the higher order iterative method introduced by Srivastava and Gupta in [26].

Now, using only the Penrose Equation (2.4b) and for arbitrary $\beta \in \mathbb{R}$, we get

$$X = X + \beta(2X - 3XAX + XAXAX),$$

or equivalently

$$X = X + \beta X(2I - 3AX + (AX)^2).$$

This leads to the following third order method

$$X_{k+1} = X_k + \beta X_k(2I - 3AX_k + (AX_k)^2), \quad k = 0, 1, \dots \quad (3.9)$$

with

$$X_0 = \alpha A^*,$$

where α is an appropriate real number.

The method (3.9) appeared also in [9].

This can further be extended to the p -th order for $p \geq 3$, given by

$$X_{k+1} = X_k + \beta X_k ((I - AX_k) + (I - AX_k)^2 + \cdots + (I - AX_k)^{p-1}), \quad k = 0, 1, \dots \quad (3.10)$$

with

$$X_0 = \alpha A^*,$$

where α is an appropriate real number.

Lemma 3.2.1. *For all $k \geq 0$, the sequence X_k generated by (3.9) or (3.10) satisfies:*

$$(X_k A)^* = (X_k A), \quad (3.11a)$$

$$X_k A X = X_k, \quad (3.11b)$$

$$X A X_k = X_k. \quad (3.11c)$$

Proof. We use mathematical induction. For $k = 0$ we have $X_0 = \alpha A^*$ and all statements in (3.11) hold. Assume the statements are true for some integer k . To show that it also holds for $k + 1$, we consider

$$\begin{aligned} (X_{k+1} A)^* &= (X_k A)^* + \beta \{2(X_k A)^* - 3((X_k A)^*)^2 + ((X_k A)^*)^3\} \\ &= X_k A + \beta (2X_k A - 3(X_k A)^2 + (X_k A)^3) \\ &= X_k A + \beta (2X_k - 3X_k A X_k + X_k (A X_k)^2) A \\ &= X_{k+1} A. \end{aligned}$$

Hence it holds for all $k \geq 0$. Likewise for p -th order method (3.10), we get

$$(X_{k+1} A)^* = \{X_k A + \beta X_k ((I - AX_k) + \cdots + (I - AX_k)^{p-1}) A\}^*$$

$$\begin{aligned}
 &= (X_k A)^* + \beta((X_k(I - AX_k)A)^* + \cdots + (X_k(I - AX_k)^{p-1}A)^*) \\
 &= (X_k A) + \beta((X_k(I - AX_k)A) + \cdots + (X_k(I - AX_k)^{p-1}A)) \\
 &= X_k A + \beta X_k((I - AX_k) + \cdots + (I - AX_k)^{p-1})A \\
 &= X_{k+1}A,
 \end{aligned}$$

note that, we prove

$$(X_k(I - AX_k)^p A)^* = X_k(I - AX_k)^p A, \quad p \geq 1$$

by mathematical induction. The statement is clear when $p = 1$. Assume that it is true for a fixed integer $p \geq 1$. Then

$$\begin{aligned}
 (X_k(I - AX_k)^{p+1}A)^* &= (X_k(I - AX_k)^p(I - AX_k)A)^* \\
 &= (X_k(I - AX_k)^p A(I - X_k A))^* \\
 &= (I - X_k A)^*(X_k(I - AX_k)^p A)^* \\
 &= (I - X_k A)(X_k(I - AX_k)^p A) \\
 &= (X_k - X_k A X_k)(I - AX_k)^p A \\
 &= X_k(I - AX_k)^{p+1}A.
 \end{aligned}$$

Now, for (3.11b)

$$\begin{aligned}
 X_{k+1}AX &= X_k AX + \beta X_k(2I - 3AX_k + (AX_k)^2)AX \\
 &= X_k + \beta X_k(2I - 3AX_k + (AX_k)^2) \\
 &= X_{k+1}.
 \end{aligned}$$

Likewise for p -th order method (3.10) with the fact that $(AX)^p = (AX)$, we get

$$\begin{aligned}
 X_{k+1}AX &= X_k AX + \beta X_k((I - AX_k) + (I - AX_k)^2 + \cdots + (I - AX_k)^{p-1})AX \\
 &= X_k + \beta X_k((I - AX_k)AX + (I - AX_k)^2(AX)^2 + \cdots + (I - AX_k)^{p-1}(AX)^{p-1})
 \end{aligned}$$

$$\begin{aligned}
 &= X_k + \beta X_k((AX - AX_kAX) + (AX - AX_kAX)^2 + \cdots + (AX - AX_kAX)^{p-1}) \\
 &= X_k + \beta X_k(AX(I - AX_k) + (AX)^2(I - AX_k)^2 + \cdots + (AX)^{p-1}(I - AX_k)^{p-1}) \\
 &= X_k + \beta X_kAX((I - AX_k) + (I - AX_k)^2 + \cdots + (I - AX_k)^{p-1}) \\
 &= X_k + \beta X_k((I - AX_k) + (I - AX_k)^2 + \cdots + (I - AX_k)^{p-1}) \\
 &= X_{k+1}.
 \end{aligned}$$

For (3.11c), we have

$$\begin{aligned}
 XAX_{k+1} &= XAX_k + \beta XAX_k(2I - 3AX_k + (AX_k)^2) \\
 &= X_k + \beta X_k(2I - 3AX_k + (AX_k)^2) \\
 &= X_{k+1}.
 \end{aligned}$$

And for p -th order method (3.10), we get

$$\begin{aligned}
 XAX_{k+1} &= XAX_k + \beta XAX_k((I - AX_k) + (I - AX_k)^2 + \cdots + (I - AX_k)^{p-1}) \\
 &= X_k + \beta X_k((I - AX_k) + (I - AX_k)^2 + \cdots + (I - AX_k)^{p-1}) \\
 &= X_{k+1}.
 \end{aligned}$$

This completes the proof of the lemma. □

Theorem 3.2.1. *Iterative method (3.9) with the initial approximation $X_0 = \alpha A^*$ converges to the Moore-Penrose inverse $X = A^\dagger$ under the assumptions*

$$\|(X_0 - X)A\| < 1, \quad 0 < \beta \leq 1. \quad (3.12)$$

It has linear convergence for $\beta < 1$ and third order convergence for $\beta = 1$.

Proof. Using Lemma 3.2.1 and substituting for X_k , we get

$$\|E_k\| = \|X_k - X\| = \|X_kAX - XAX\| \leq \|X_kA - XA\| \|X\| = \|t_k\| \|X\|,$$

where $t_k = X_k A - X A = E_k A$.

Again using Lemma 3.2.1 with (3.9), we get

$$\begin{aligned}
 t_{k+1} &= X_{k+1} A - X A = (X_k + \beta X_k (2I - 3A X_k + (A X_k)^2)) A - X A \\
 &= X_k A - X A + \beta (2X_k A - 3(X_k A)^2 + (X_k A)^3) \\
 &= X_k A - X A + \beta ((X_k A)^3 - X A + 3(X_k A) - 3(X_k A)^2 - (X_k A - X A)) \\
 &= X_k A - X A + \beta ((X_k A - X A)^3 - (X_k A - X A)) \\
 &= \beta t_k^3 + (1 - \beta) t_k. \tag{3.13}
 \end{aligned}$$

Let $s_k = \|t_k\|$, we require that $s_k \rightarrow 0$ as $k \rightarrow \infty$. Condition (3.12) implies $s_0 < 1$, then by an assumption that $s_k < 1$, from (3.13) and inductive method, we obtain

$$s_{k+1} \leq \beta s_k^3 + (1 - \beta) s_k < \beta s_k + (1 - \beta) s_k < s_k < 1. \tag{3.14}$$

Thus, s_k is a monotonically decreasing bounded sequence converging to s as $k \rightarrow \infty$ and $0 \leq s < 1$. From (3.14), we get

$$s \leq \beta s^3 + (1 - \beta) s,$$

then

$$\beta s^3 - \beta s \geq 0.$$

This gives either $s = 0$ or $s \geq 1$. Thus, $s = 0$.

This completes the proof that $s_k \rightarrow 0$ as $k \rightarrow \infty$. Thus, $X_k \rightarrow X$ as $k \rightarrow \infty$.

From (3.13), we conclude that iterative method (3.9) has linear convergence if $\beta < 1$, and its convergence is third if $\beta = 1$. \square

In general, we have the following theorem.

Theorem 3.2.2. *Iterative method (3.10) with the initial approximation $X_0 = \alpha A^*$ con-*

verges to the Moore-Penrose inverse $X = A^\dagger$ under the assumptions

$$\|(X_0 - X)A\| < 1, \quad 0 < \beta \leq 1. \quad (3.15)$$

It has linear convergence for $\beta < 1$ and p -th order convergence for $\beta = 1$, where, $p \geq 2$ is a positive integer.

Next, we consider some numerical tests.

Example 3.3. Consider the matrix A of order (5×6) given by

$$A = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 4 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 1 & 2 & -2 & 0 & -3 \end{bmatrix}.$$

The choice $\alpha = 0.0185$ satisfies the convergence criterion given by

$$\max_{1 \leq i \leq 5} |1 - \alpha \lambda_i(A^*A)| = 0.9988 < 1,$$

since the eigenvalues of the matrix A^*A are

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = (0.0623, 1.2229, 4.7108, 15.9912, 23.0128).$$

The iterative method (3.10) with tolerance $\epsilon = 10^{-8}$ converges to the Moore-Penrose inverse A^\dagger in 14 iteration for $p = 2$ and with 9 iterations for $p = 3$.

$$A^\dagger = \begin{bmatrix} -0.1439 & 0.3030 & -0.1364 & -0.1894 & 0.0758 \\ -1.1515 & 0.4242 & -0.0909 & -1.5152 & 0.6061 \\ 1.2803 & -0.4848 & 0.3182 & 2.0530 & -0.6212 \\ -1.1439 & 0.3030 & -0.1384 & -0.1894 & 0.0758 \\ -0.5758 & 0.2121 & -0.5455 & -0.7576 & 0.3030 \\ 1.2803 & -0.4848 & 0.3182 & 1.0530 & -0.6212 \end{bmatrix}.$$

Example 3.4. Consider the ill-conditional Hilbert matrix A of order (5×5) appeared in Example 3.2.

The iterative method (3.10) generates a sequence of iterates $\{X_k\}$ converging to the Moore-Penrose inverse A^\dagger given by

$$A^\dagger = \begin{bmatrix} 25 & -300 & 1050 & -1400 & 630 \\ -300 & 4800 & -18900 & 26880 & -12600 \\ 1050 & -18900 & 79380 & -117600 & 56700 \\ -1400 & 26880 & -117600 & 179200 & -88200 \\ 630 & -12600 & 56700 & -88200 & 44100 \end{bmatrix}.$$

A comparison for number of iterations versus the order are plotted in Figure 3.1. It can be observed that the iterative method (3.10) converges to the Moore-Penrose inverse A^\dagger in 42 iterations for $p = 2$ and as the order p increases, it reduces to 14 for $p = 10$. We note that after $p = 7$ the number of iteration still fixed.

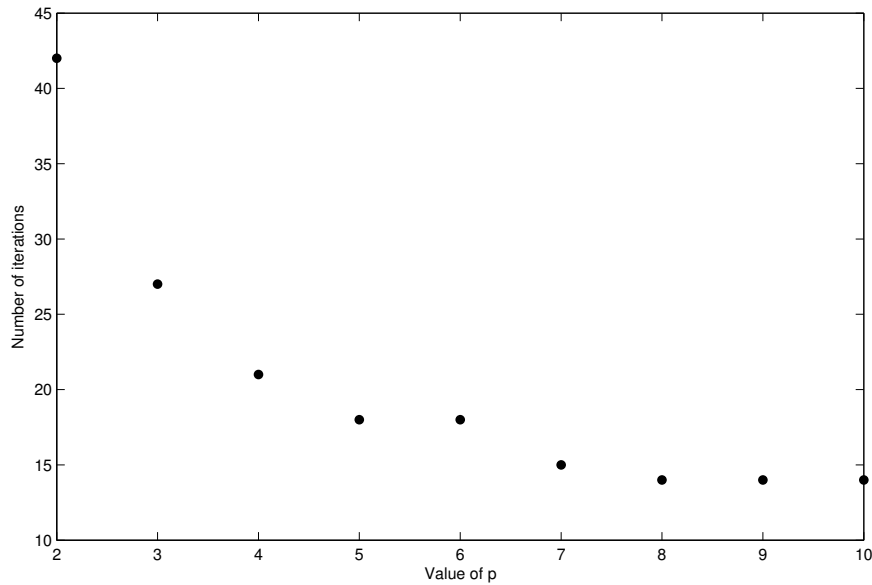


Figure 3.1: Number of iterations versus the value of p , Example 3.4.

Example 3.5. Let $A = \text{rand}(50,50)$, we have tested 50 times with MATLAB. The number of iterations and average of CPU time required for convergence are compared between Newton method and third order method (3.9) in figures 3.2 and 3.3, respectively.

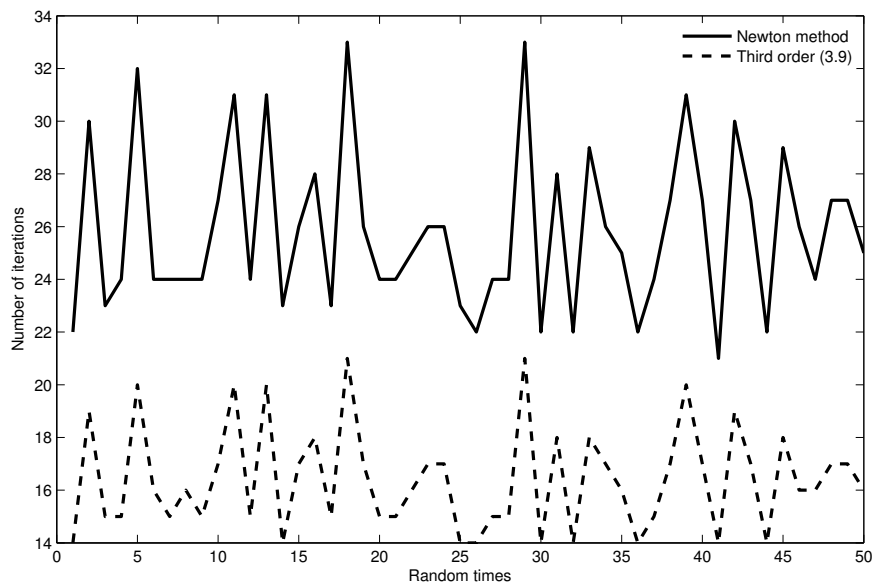


Figure 3.2: Comparison number of iterations, Example 3.5.

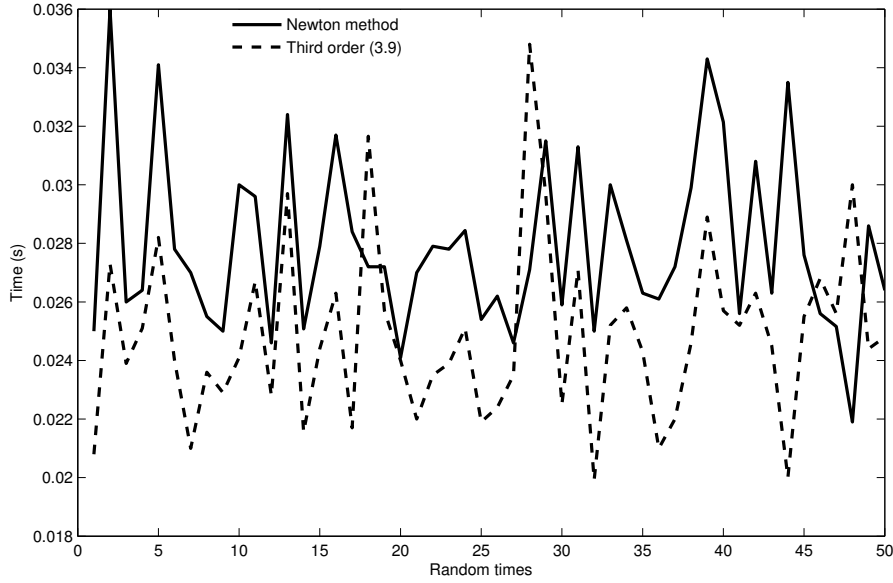


Figure 3.3: The results of comparisons of computational time, Example 3.5.

3.3 Fourth order iterative method

In this section, we present fourth order iterative method for computing Moore-Penrose inverse. Convergence analysis is considered.

Esmaeili et. al. in [5] proposed the following fourth-order iterative method

$$X_{k+1} = X_k[9I - 26(AX_k) + 34(AX_k)^2 - 21(AX_k)^3 + 5(AX_k)^4], \quad k = 0, 1, 2, \dots, \quad (3.16)$$

with

$$X_0 = \alpha A^*,$$

where α is an appropriate real number.

Let us consider the following singular value decomposition of the matrix $A \in \mathbb{C}^{m \times n}$ of $\text{rank}(A) = r \leq \min\{m, n\}$

$$A = V \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} U^*, \quad S = \text{diag}(\sigma_1, \dots, \sigma_r), \quad \sigma_1 \geq \dots \geq \sigma_r > 0.$$

The Moore-Penrose inverse is given by

$$A^\dagger = U \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^*.$$

Where U and V are unitary matrices. Using

$$X_0 = \alpha A^*,$$

in which α is a constant, we can deduce that each iterate of the method (3.16) has a singular value decomposition of the form

$$X_k = US_kV^*, \quad S_k = \text{diag}(s_1^{(k)}, \dots, s_r^{(k)}),$$

where

$$S_0 = \alpha S,$$

we have

$$US_{k+1}V^* = US_kV^*[9I - 26(VSS_KV^*) + 34(VSS_KV^*)^2 - 21(VSS_KV^*)^3 + 5(VSS_KV^*)^4].$$

Hence,

$$S_{k+1} = S_k[9I - 26SS_K + 34(SS_K)^2 - 21(SS_K)^3 + 5(SS_K)^4].$$

Therefore, the diagonal matrices $R_k = SS_K = \text{diag}(r_1^{(k)}, \dots, r_r^{(k)})$ satisfy

$$R_{k+1} = g(R_k) = 9R_k - 26R_k^2 + 34R_k^3 - 21R_k^4 + 5R_k^5,$$

that means

$$r_i^{(k+1)} = g(r_i^{(k)}), \tag{3.17}$$

where $g(r) = 9r - 26r^2 + 34r^3 - 21r^4 + 5r^5$.

In the following theorem, we show that the sequences (3.17) are fourth-order convergent to $r_i = 1$ for any $r_i^{(0)} \in (0, 1 + \gamma)$, in which γ is a suitable constant.

Theorem 3.3.1. *For any initial point $r^{(0)} \in (0, 1 + \gamma)$, the sequence $r^{(k+1)} = g(r^{(k)})$ is fourth-order convergent to $r = 1$, in which the function $g(r)$ is defined by (3.17) and $\gamma \approx 0.53$.*

Proof. The real fixed points and the critical points of $g(r)$ as follows

$$g(r) = r \Rightarrow r = 0, 1, 1 + \gamma,$$

$$g'(r) = 0 \Rightarrow r = 0.36, 1, 1, 1,$$

in which

$$\gamma = \frac{1}{15} \left[1 + \sqrt[3]{316 + 30\sqrt{114}} - \frac{14}{\sqrt[3]{316 + 30\sqrt{114}}} \right] \approx 0.53.$$

Now, $g''(0.36) = -6.55 < 0$ and $g^{(4)}(1) = 96 > 0$, we can deduce that 0.36 is a local maximizer and 1 is a local minimizer of $g(r)$.

On the other hand, $g(0) = 0 < 1 = g(1)$ and $g(0.36) \approx 1.13 < 1 + \gamma = g(1 + \gamma)$. Therefore, $r = 0, 1$ and $r = 0.36, 1 + \gamma$ are minimizer and maximizer of $g(r)$ in the interval $[0, 1 + \gamma]$, respectively.

Moreover, the interval $[0, 1 + \gamma]$ maps into itself by the function $g(r)$.

For any arbitrary $r^{(0)} \in (0, 1 + \gamma)$, we obtain

- The unique solution of the equation $g(r) = 1$ in the interval $[0, 1]$ is $\frac{1}{5}$.
- $g(r)$ increasing in the interval $\left(0, \frac{1}{5}\right)$. Therefore, if $r^{(k)} \in \left(0, \frac{1}{5}\right)$, for some k , then $r^{(k+1)} \in \left[\frac{1}{5}, 1\right]$.
- If $r^{(k)} \in \left(\frac{1}{5}, 1\right)$, for some k , then $r^{(k+1)} \in \left(1, 1 + \gamma\right)$.
- If $r^{(k)} \in \left(1, 1 + \gamma\right)$, for some k , then the sequence $\left\{r^{(r+l)}\right\}_{l \geq 1} \subseteq [1, 1 + \gamma)$ is strictly decreasing sequence converging to 1, See Figure 3.4.

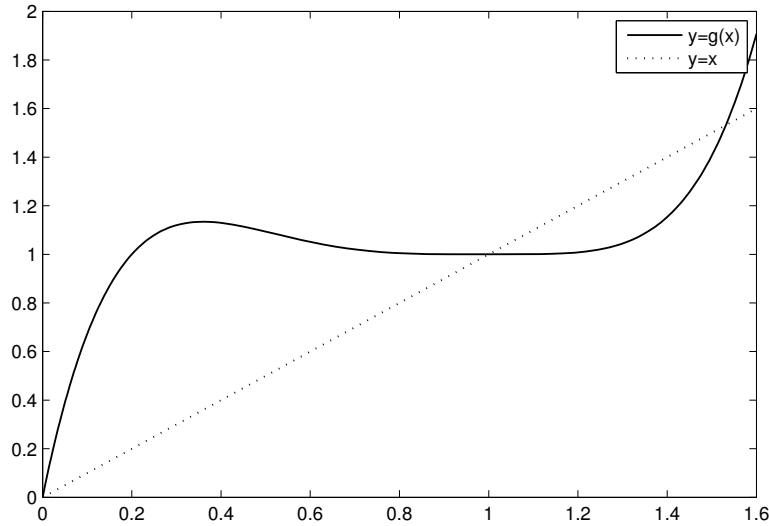


Figure 3.4: Graphs of the line $y = x$ and the function $y = g(x)$.

We can conclude that the sequence $r^{(k+1)} = g(r^{(k)})$ is convergent to $r = 1$. On the other hand,

$$g'(1) = g''(1) = g^{(3)}(1) = 0,$$

implies that the convergence is fourth order, see Lemma 2.5.4. \square

By using Theorem 3.3.1, we conclude that if $\alpha\sigma_1^2 = r_1^{(0)} \in (0, 1.53)$, then $\alpha\sigma_i^2 = r_i^{(0)} \in (0, 1.53)$, for all i , and

$$\lim_{k \rightarrow \infty} R_k = I.$$

Hence,

$$\lim_{k \rightarrow \infty} S_k = S^{-1},$$

so

$$\lim_{k \rightarrow \infty} X_k = U \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^* = A^\dagger.$$

Moreover, the order of convergence is four. Therefore, the following theorem is proved.

Theorem 3.3.2. *Consider the matrix A , and suppose that σ_1^2 denotes the largest singular value of A . Moreover, assume that the initial approximation X_0 is defined by $X_0 = \alpha A^*$,*

in which

$$0 < \alpha < \frac{1.53}{\sigma_1^2}.$$

Then, the sequence $\{X_k\}_{k \geq 0}$ generated by (3.16) converges to the Moore-Penrose inverse A^\dagger with fourth-order.

Example 3.6. Let $A = \text{rand}(50, 50)$, we have tested 50 times with MATLAB. The number of iterations and average of CPU time required for convergence are compared between fourth order method in (3.16) and fourth order reduced from (3.10) when $p = 4$ in figures 3.5 and 3.6, respectively.

We observed that from figures 3.5 and 3.6 the fourth order method (3.16) is better than the fourth order reduced from (3.10) in terms of number of iterations and CPU time.

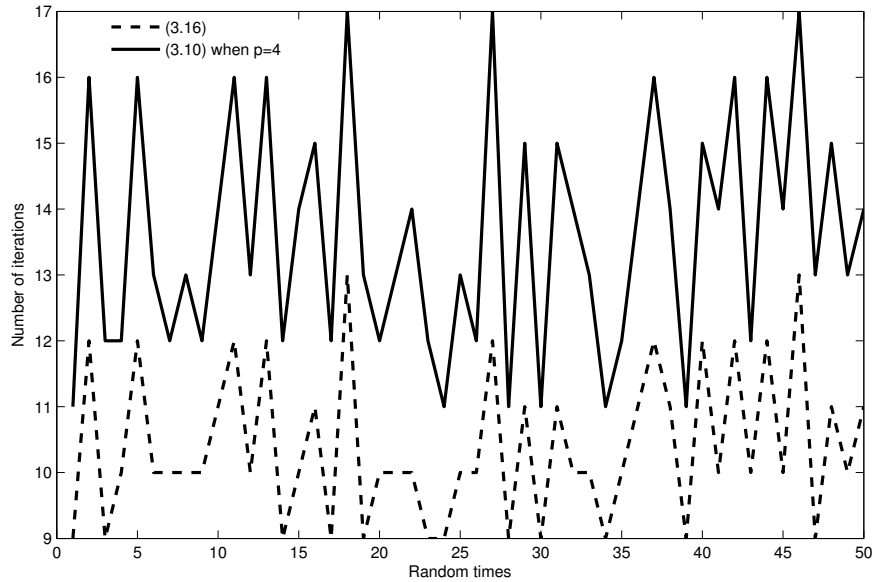


Figure 3.5: Comparison number of iteration, Example 3.6.

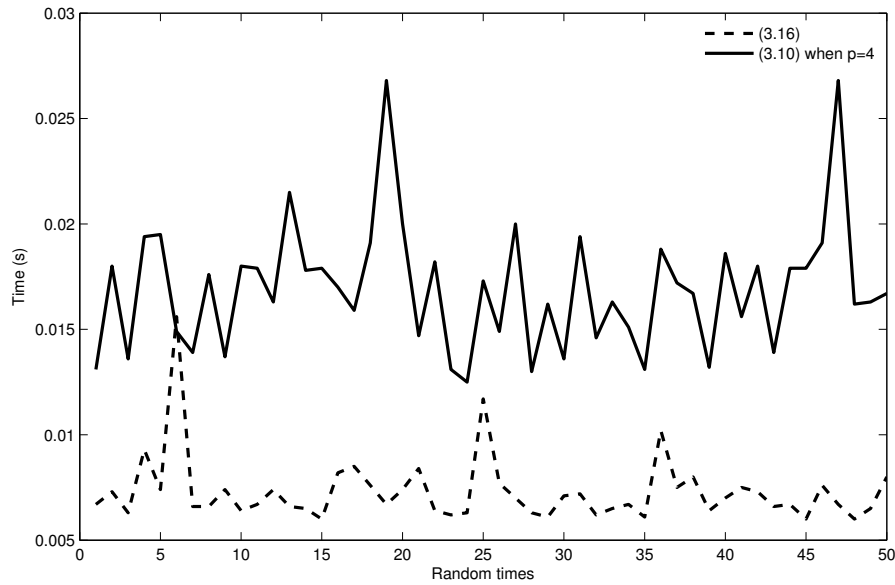


Figure 3.6: The results of comparisons of computational time, Example 3.6.

Example 3.7. Let $A = \text{rand}(50,50)$, we have tested 50 times with MATLAB. The number of iterations and average of CPU time required for convergence are compared between fourth order method in (3.16), Newton and third order method in figures 3.7 and 3.8, respectively.

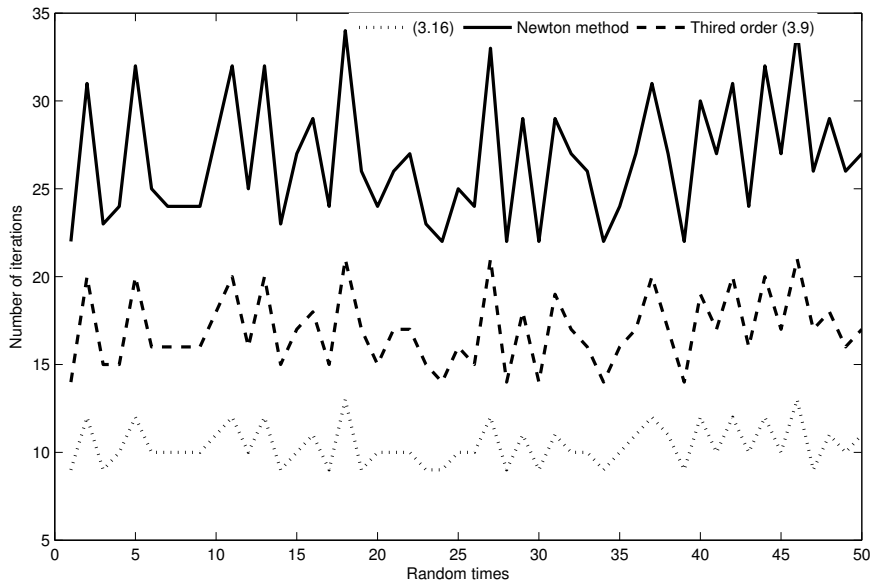


Figure 3.7: Comparison number of iteration, Example 3.7.

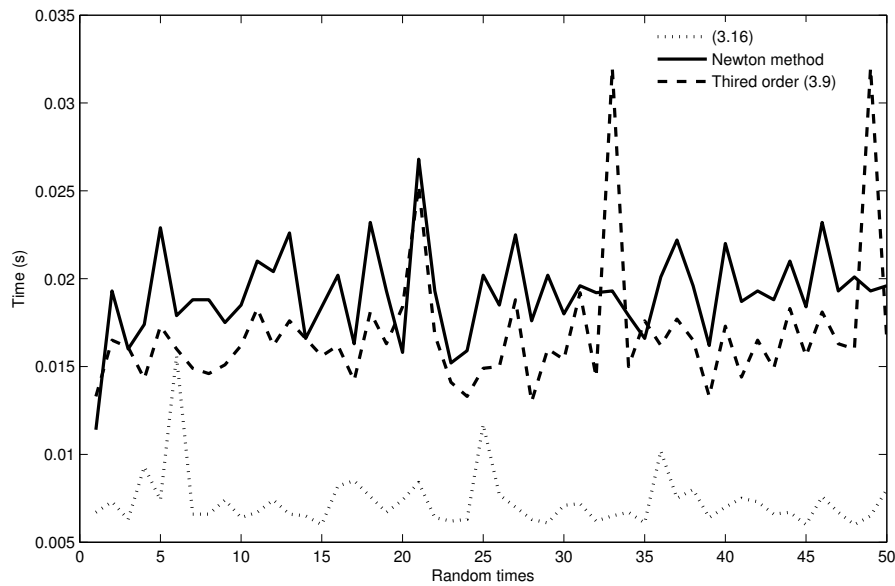


Figure 3.8: The results of comparisons of computational time, Example 3.7.

3.4 Other iterative methods

Krishnamurthy and Sen [8] provided the following fourth-order method that contains 4 matrix multiplications

$$\begin{aligned} Y_k &= I - AX_k, \\ X_{k+1} &= X_k(I + Y_k(I + Y_k(I + Y_k))). \end{aligned} \quad (3.18)$$

Solymani et. al [25] provided the following sixth-order method that contains 5 matrix multiplications

$$\begin{aligned} B_k &= AX_k \\ S_k &= B_k(-I + B_k) \\ X_{k+1} &= X_k(2I - B_k)(3I - 2B_k + S_k)(I + S_k) \end{aligned} \quad (3.19)$$

Toutounian and Soleymani [28] proposed the following fourth-order method that involves 5 matrix multiplications

$$X_{k+1} = 0.5X_k(9I - AX_k(16I - AX_k(14I - AX_k(6I - AX_k)))). \quad (3.20)$$

Soleymani et. al. [23] presented the following ninth-order method that has 7 matrix multiplications in each iteration

$$\begin{aligned} B_k &= AX_k \\ S_k &= 3I + B_k(-3I + B_k) \\ T_k &= B_k S_k \\ X_{k+1} &= -\frac{1}{4}X_k S_k(13I + T_k(15I + T_k(-7I + T_k))) \end{aligned} \quad (3.21)$$

Sharifi et. al. [22] proposed the following 30-order method that has only 9 matrix multiplications in each iteration

$$\begin{aligned} R_k &= I - AX_k \\ X_{k+1} &= X_k(I + R_k)(I + R_k^2 + R_k^4)(I + (R_k^2 + R_k^8)(R_k^4 + R_k^{16})). \end{aligned} \quad (3.22)$$

Soleimani et. al. [24] proposed the following 31-order method that has only 9 matrix multiplications in each iteration

$$\begin{aligned} R_k &= I - AX_k \\ X_{k+1} &= X_k(I + (R_k + R_k^2)(I + R_k^2 + R_k^4)(I + (R_k^2 + R_k^8)(R_k^4 + R_k^{16}))). \end{aligned} \quad (3.23)$$

3.5 The choices of the initial value X_0

The choices for the initial value X_0 on iterative (3.1), (3.9) and (3.10) are very important to preserve convergence. Recently, there exist many different forms for the initial value X_0 . In this work, we use the initial value as in [14].

For a square $m \times m$ matrix A , Rajagopalan in [16] constructed the initial value as

$$X_0 = \frac{A^T}{m\|A\|_1\|A\|_\infty},$$

or $X_0 = \alpha I$, where I is the identity matrix and $\alpha \in \mathbb{R}$ should adaptively be determined such that $\|I - \alpha A\| < 1$.

For diagonally dominant matrices, Sciavico and Siciliano in [19] used

$$X_0 = \text{diag}\left(\frac{1}{a_{11}}, \frac{1}{a_{22}}, \dots, \frac{1}{a_{mm}}\right),$$

where a_{ii} is the i -th diagonal entry of A , $i = 1, 2, \dots, m$.

For a symmetric positive definite matrix, Codevico et.al. in [4] used

$$X_0 = \frac{1}{\|A\|_F} I,$$

where $\|\cdot\|_F$ is the Frobenius norm.

For rectangular or singular matrices, one may choose

$$X_0 = \frac{A^*}{(\|A\|_1\|A\|_\infty)} \text{ or } X_0 = \frac{A^T}{(\|A\|_1\|A\|_\infty)},$$

based on [14].

We could choose initial value as in [1]

$$X_0 = \alpha A^*,$$

where $0 < \alpha < \frac{2}{\rho(A^*A)}$.

Chapter 4

Our new iterative methods

In this chapter, we present our methods for computing the Moore-Penrose inverse. The construction of these algorithm is based on the usage of Penrose Equations (2.4a) and (2.4b). Two different schemes are established. And convergence properties are considered. Numerical result is also presented and a comparison with Newton's method is made.

In Section 4.1, we establish a new family of second order iterative methods. These methods are written interms of $p - th$ root of a matrix AX_k . In Section 4.2, a second order iterative method uses $(AX_k)^2$ is constructed.

4.1 A new family of second-order iterative methods

In this section, we establish a fast and efficient new family of second-order iterative algorithms for computing the Moore-Penrose inverse.

Let $A \in \mathbb{C}^{m \times n}$ and $X = A^\dagger \in \mathbb{C}^{n \times m}$. We use Equations (2.4a) and (2.4b) to obtain

$$X = XAX = X(AXAX)^{\frac{1}{2}} = X(AX)^{\frac{1}{2}}.$$

Hence, we have

$$X = X - 2(X(AX)^{\frac{1}{2}} - X).$$

From the last equation we get the following iterative method

$$X_{k+1} = X_k - 2X_k((AX_k)^{\frac{1}{2}} - I). \tag{4.1}$$

Assume the starting value for the iterative method (4.1) is

$$X_0 = \alpha A^*, \quad (4.2)$$

for an appropriate real number α .

Continuing in a similar manner, this can further be extended to a family of second-order iterative method, given by

$$X_{k+1} = X_k - pX_k((AX_k)^{\frac{1}{p}} - I), \quad p \in \{2, 3, 4, \dots\}. \quad (4.3)$$

Lemma 4.1.1. *The iterative schemes (4.1) and (4.3) satisfy the following relations*

$$XAX_k = X_k, \quad (4.4a)$$

$$X_kAX = X_k, \quad (4.4b)$$

where $k \geq 0$.

Proof. We use mathematical induction. For $k = 0$ we have $X_0 = \alpha A^*$ and all statements in (4.4) hold. Assume the statements are true for some integer k . Now we prove the statements for $k + 1$.

For (4.4a), we have

$$\begin{aligned} XAX_{k+1} &= XA(X_k - 2X_k((AX_k)^{\frac{1}{2}} - I)) \\ &= XAX_k - 2XAX_k((AX_k)^{\frac{1}{2}} - I) \\ &= X_k - 2X_k((AX_k)^{\frac{1}{2}} - I) \\ &= X_{k+1}. \end{aligned}$$

we prove (4.4b) in a similar way

$$X_{k+1}AX = (X_k - 2X_k((AX_k)^{\frac{1}{2}} - I))AX$$

$$\begin{aligned}
 &= X_k AX - 2(X_k (AX_k)^{\frac{1}{2}} AX - X_k AX) \\
 &= X_k - 2(X_k (AX_k)^{\frac{1}{2}} (AX)^{\frac{1}{2}} - X_k) \\
 &= X_k - 2(X_k (AX_k AX)^{\frac{1}{2}} - X_k) \\
 &= X_k - 2(X_k (AX_k)^{\frac{1}{2}} - X_k) \\
 &= X_{k+1}.
 \end{aligned}$$

Proceeding in a similar manner, (4.4) can easily be proved for (4.3).

This completes the proof of the lemma. □

Now, we follow the idea of [5] to prove that the matrix sequence X_k defined by the iterative method (4.1) and the starting value (4.2), converges to the Moore-Penrose inverse $X = A^\dagger$.

Let us consider the following singular value decomposition of the matrix $A \in \mathbb{C}^{m \times n}$ of $\text{rank}(A) = r \leq \min\{m, n\}$

$$A = V \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} U^*, \quad S = \text{diag}(\sigma_1, \dots, \sigma_r), \quad \sigma_1 \geq \dots \geq \sigma_r > 0.$$

Where σ_i are the singular values of A .

The Moore-Penrose is given by

$$A^\dagger = U \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^*.$$

Where U and V are unitary matrices. Using

$$X_0 = \alpha A^*,$$

in which α is a constant, we can deduce that each iterate of the method (4.1) has a singular value decomposition of the form

$$X_k = U S_k V^*, \quad S_k = \text{diag}(s_1^{(k)}, \dots, s_r^{(k)}),$$

where

$$S_0 = \alpha S,$$

we have

$$\begin{aligned} US_{k+1}V^* &= US_kV^* - 2US_kV^*((VSS_kV^*)^{\frac{1}{2}} - I) \\ &= 3US_kV^* - 2US_kV^*(VSS_kV^*)^{\frac{1}{2}} \\ &= 3US_kV^* - 2US_kV^*V(SS_k)^{\frac{1}{2}}V^*. \end{aligned}$$

Hence,

$$S_{k+1} = 3S_k - 2S_k(SS_k)^{\frac{1}{2}}.$$

Therefore, the diagonal matrices $R_k = SS_k = \text{diag}(r_1^{(k)}, \dots, r_r^{(k)})$ satisfy

$$R_{k+1} = g(R_k) = 3R_k - 2R_k(R_k)^{\frac{1}{2}},$$

that means

$$r_i^{(k+1)} = g(r_i^{(k)}) = 3r_i^{(k)} - 2r_i^{(k)\frac{3}{2}}. \quad (4.5)$$

In general, for (4.3) we have

$$S_{k+1} = (p+1)S_k - pS_k(SS_k)^{\frac{1}{p}}.$$

Therefore, the diagonal matrices $R_k = SS_k = \text{diag}(r_1^{(k)}, \dots, r_r^{(k)})$ satisfy

$$R_{k+1} = g(R_k) = (p+1)R_k - pR_k(R_k)^{\frac{1}{p}},$$

hence

$$r_i^{(k+1)} = g(r_i^{(k)}) = (p+1)r_i^{(k)} - pr_i^{(k)\frac{p+1}{p}}. \quad (4.6)$$

Theorem 4.1.1. *For any initial point $r^{(0)} \in \left(0, \frac{16}{9}\right)$, the sequence $r^{(k+1)} = g(r^{(k)})$ is of a second order convergence to $r = 1$, in which the function $g(r)$ is defined by (4.5).*

Proof. The fixed points and the critical points of $g(r)$ are

$$g(r) = r \Rightarrow r = 0, 1,$$

$$g'(r) = 0 \Rightarrow r = 1.$$

We can find that 1 is local maximizer of $g(r)$. It is easy to see that the interval $\left(\frac{4}{9}, \frac{16}{9}\right)$ is mapped into itself.

Moreover, $g(r)$ is a continuous function on the interval $\left(\frac{4}{9}, \frac{16}{9}\right)$, and $|g'(r)| < 1$ on this interval.

We conclude that the sequence $r^{(k+1)} = g(r^{(k)})$ is convergent to $r = 1$, by Lemma 2.5.3. For the interval $\left(0, \frac{4}{9}\right]$ the sequence $r^{(k+1)} = g(r^{(k)}) > r^{(k)}$, increasing and bounded above, see Figure 4.1. Hence we obtain convergent for any $r^{(0)} \in \left(0, \frac{16}{9}\right)$. On the other hand,

$$g(1) = 1, \quad g'(1) = 0,$$

implies that the convergence is second order, by Lemma 2.5.4. \square

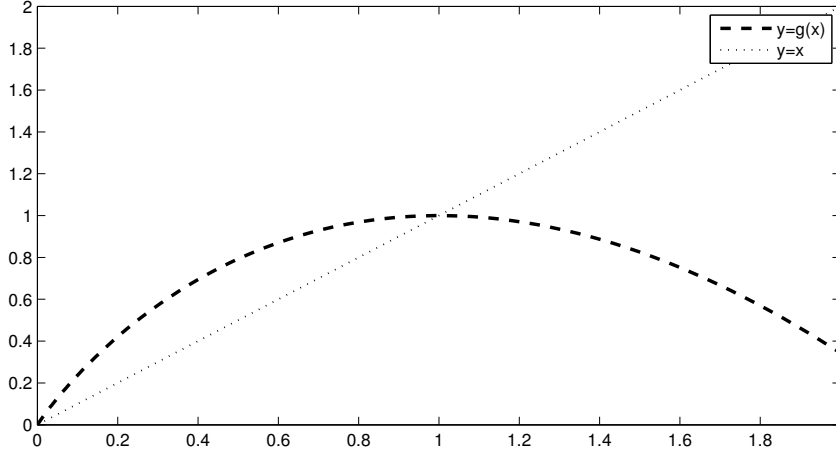


Figure 4.1: Graph of the function $y = g(x)$ and the line $y = x$.

Considering Theorem 4.1.1, we conclude that if $\alpha\sigma_1^2 = r_1^{(0)} \in \left(0, \frac{16}{9}\right)$, then $\alpha\sigma_i^2 = r_i^{(0)} \in \left(0, \frac{16}{9}\right)$, for all i , and

$$\lim_{k \rightarrow \infty} R_k = I.$$

Hence,

$$\lim_{k \rightarrow \infty} S_k = S^{-1},$$

so

$$\lim_{k \rightarrow \infty} X_k = U \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^* = A^\dagger.$$

Hence, the following theorem is proved.

Theorem 4.1.2. *Let A be an $m \times n$ nonzero complex matrix. If the initial approximation X_0 is defined by*

$$X_0 = \alpha A^*, \quad \text{with } 0 < \alpha < \frac{\frac{16}{9}}{\sigma_1^2}, \quad (4.7)$$

then the iterative method (4.1) converges to A^\dagger with second order, where σ_1^2 denotes the largest singular value of A .

In addition, we have the following theorem.

Theorem 4.1.3. *Let A be an $m \times n$ nonzero complex matrix. If the initial approximation X_0 is defined by*

$$X_0 = \alpha A^*, \quad \text{with } 0 < \alpha < \frac{\frac{16}{9}}{\sigma_1^2}, \quad (4.8)$$

then

$$\|A(X - X_0)\| < 1.$$

Proof. Take $P = AX$ and $Q = I - AX_0$. Since $P^2 = P$ and

$$\begin{aligned} PQ &= AX - AXAX_0 = AX - AX_0 \\ &= AX - AX_0AX \\ &= (I - AX_0)AX \\ &= QP, \end{aligned}$$

from Lemma 2.5.2 we can conclude that

$$\rho(A(X - X_0)) \leq \rho(I - AX_0) = \rho(I - \alpha AA^*) = \max_{1 \leq i \leq r} |1 - \alpha \lambda_i(AA^*)| = \max_{1 \leq i \leq r} |1 - \alpha \sigma_i^2|.$$

By using (4.8), we conclude that

$$\rho(A(X - \alpha A^*)) \leq \max_{1 \leq i \leq r} |1 - \alpha \sigma_i^2| < 1.$$

Then from Lemma 2.5.1, we have

$$\|A(X - X_0)\| < 1.$$

□

Theorem 4.1.4. For any initial point $r^{(0)} \in \left(0, \left(\frac{p+2}{p+1}\right)^p\right)$, the sequence $r^{(k+1)} = g(r^{(k)})$ is second order convergent to $r = 1$, in which the function $g(r)$ is defined by (4.6).

Proof. The proof is similar to that of Theorem 4.1.1. The general behaviour of $g(r)$ defined in (4.6) is similar to the case when $p = 2$.

See Figure 4.2 which is the graph of (4.6) when $p = 3$, $p = 6$ and $p = 12$. □

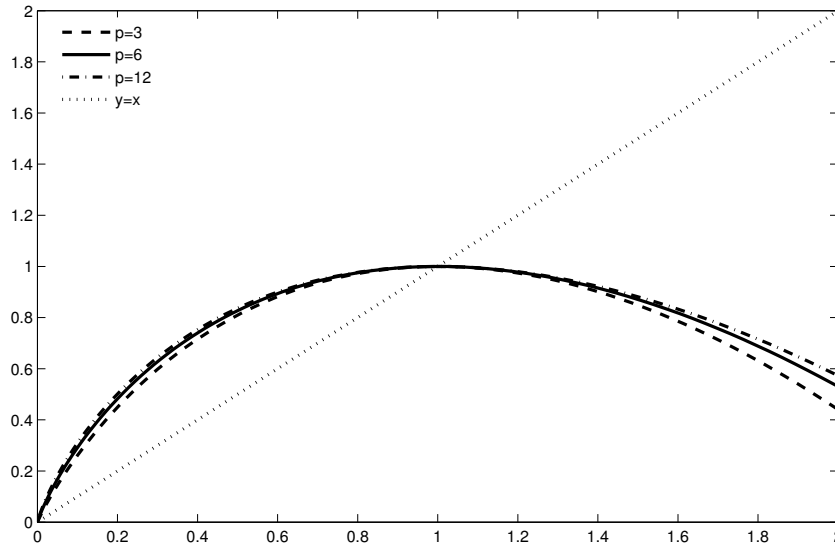


Figure 4.2: Graphs of the function $y = g(x)$ with different values of p and the line $y = x$.

Our method uses the p -th root of a square matrix AX_k . One can find several algorithms to compute this. In this work, we replace the p -th root by finite terms from power series expansion for a matrix of form $(I + B)^{\frac{1}{p}}$ which is given in the next remark.

Remark 4.1. Let $p \geq 2$ be an integer and $\|B\| < 1$, the power series expansion can be applied to define the matrix p -th root of the matrix $(I + B)$ as

$$(I + B)^{\frac{1}{p}} = \sum_{n=0}^{\infty} \frac{\frac{1}{p} \left(\frac{1}{p} - 1\right) \dots \left(\frac{1}{p} - n + 1\right)}{n!} B^n, \quad (4.9)$$

see e.g. [10].

By approximating $(AX_k)^{\frac{1}{2}}$ or $(I + (AX_k - I))^{\frac{1}{2}}$ with n terms of (4.9) we obtain the following iterative method

$$X_{k+1} = X_k - 2X_k \left(I + \frac{1}{2}(AX_k - I) - \frac{1}{8}(AX_k - I)^2 + \dots + \frac{1}{n!} \frac{1}{2} \left(\frac{1}{2} - 1\right) \dots \left(\frac{1}{2} - n + 1\right) (AX_k - I)^n - I \right). \quad (4.10)$$

With the starting value $X_0 = \alpha A^*$, where $0 < \alpha < \frac{16}{9\sigma_1^2}$, then $\rho(I - AX_0) < 1$.

In the following, several examples are given to show the efficient of our method. We use (4.10) up to $n = 2$. We find that if we use more terms from (4.10) the number of iteration decreases. But, after $n = 4$ the number of iterations is fixed.

Example 4.1. Consider the ill-conditional Hilbert matrix A of order (5×5) appeared in Example 3.2.

The iterative method (4.1) generates a sequence of iterates $\{X_k\}$ after 39 steps converging to the Moore-Penrose inverse A^\dagger given by

$$A^\dagger = \begin{bmatrix} 25 & -300 & 1050 & -1400 & 630 \\ -300 & 4800 & -18900 & 26880 & -12600 \\ 1050 & -18900 & 79380 & -117600 & 56700 \\ -1400 & 26880 & -117600 & 179200 & -88200 \\ 630 & -12600 & 56700 & -88200 & 44100 \end{bmatrix}.$$

While Newton method needs 42 iterations to have the same result.

Example 4.2. For the ill-conditional Hilbert matrix A of order (5×5) we used the iterative method (4.3) for different value of p .

It can be observed from Figure 4.3 that the iterative (4.3) converges to the Moore-Penrose inverse. The comparison of number of iterations are plotted in Figure 4.3. We note that for $p \geq 10$ the number of required iteration still fixed.

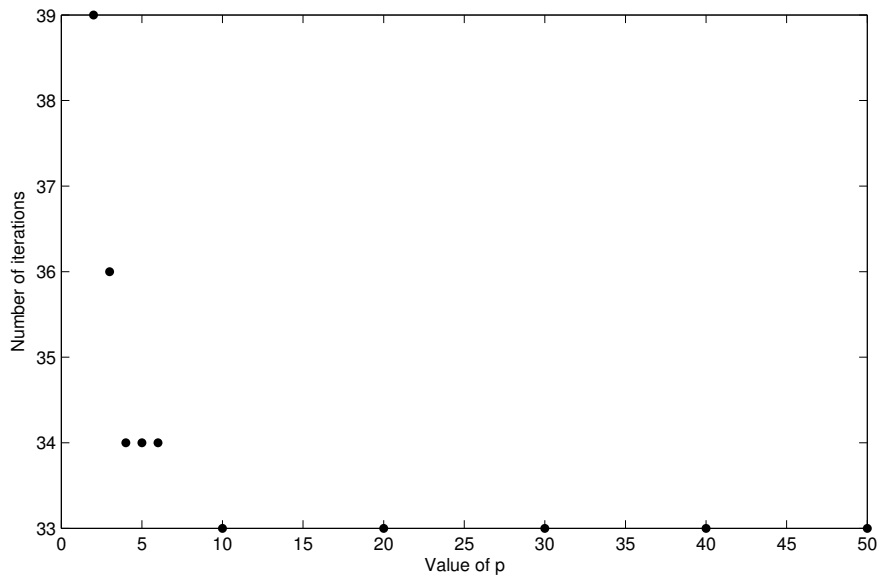


Figure 4.3: Number of iterations versus the value of p , Example 4.2.

Example 4.3. We compute the Moore-Penrose inverse random square matrix A , where A are randomly generated as follows

$$A = 20\text{rand}(600 + n, 600 + n) - 10\text{rand}(600 + n, 600 + n),$$

where n is $n = 0, 100, 200, 300, 400, \dots, 3000$.

The number of iterations and the CPU time required for convergence are compared in figures 4.4 and 4.5, respectively.

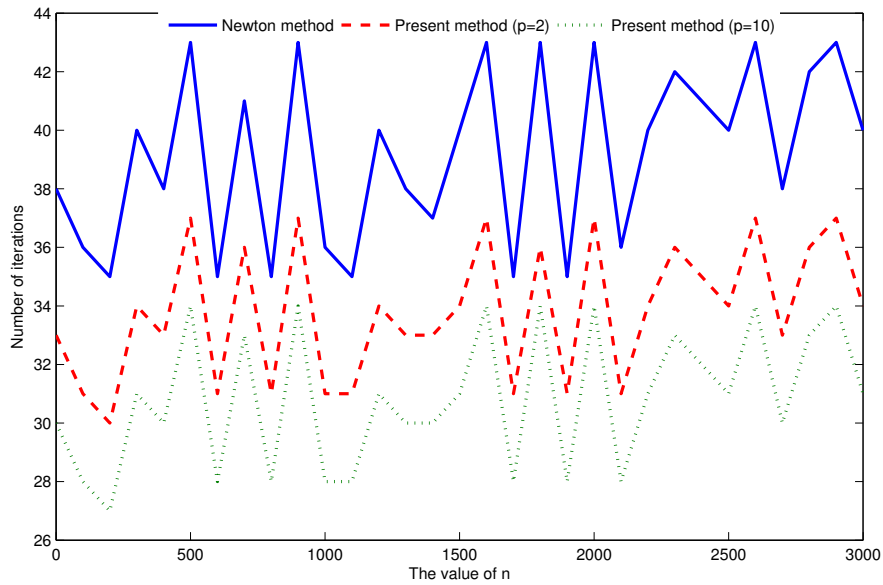


Figure 4.4: Comparison number of iteration, Example 4.3.

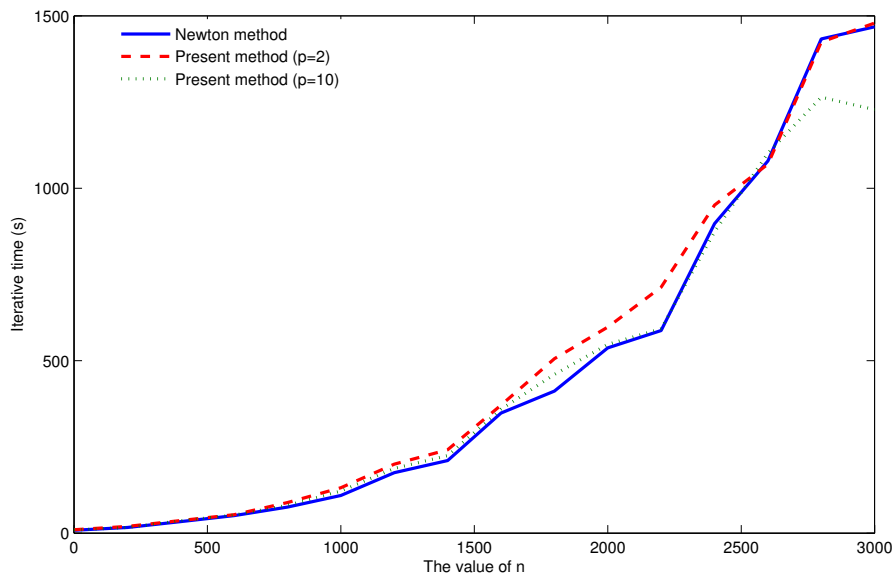


Figure 4.5: The results of comparisons of computational time, Example 4.3.

We see that the required number of iterations for the current method is less than that of Newton’s method. But the computational time is almost the same.

We noted that for the matrices $A_{m \times n}$ with $m < n$ the current methods also require less time. Next example illustrate this idea.

Example 4.4. We compute the inverse random square matrix A , where A are randomly generated as follows

$$A = 20\text{rand}(500, 1000 + n) - 10\text{rand}(500, 1000 + n),$$

and the value of n is $n = 0, 100, 200, 300, 400, \dots, 3000$.

The number of iterations and the CPU time required for convergence are compared in figures 4.6 and 4.7, respectively.

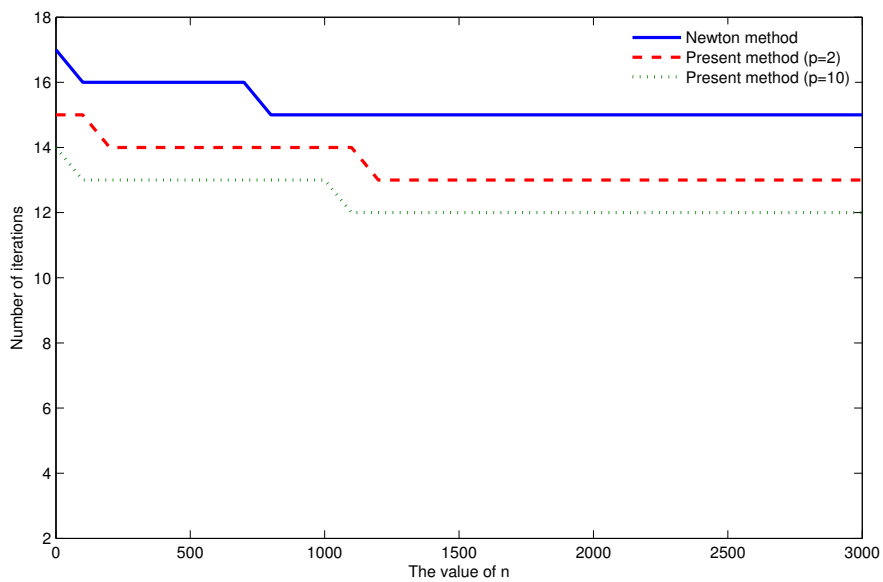


Figure 4.6: Comparison number of iteration, Example 4.4.

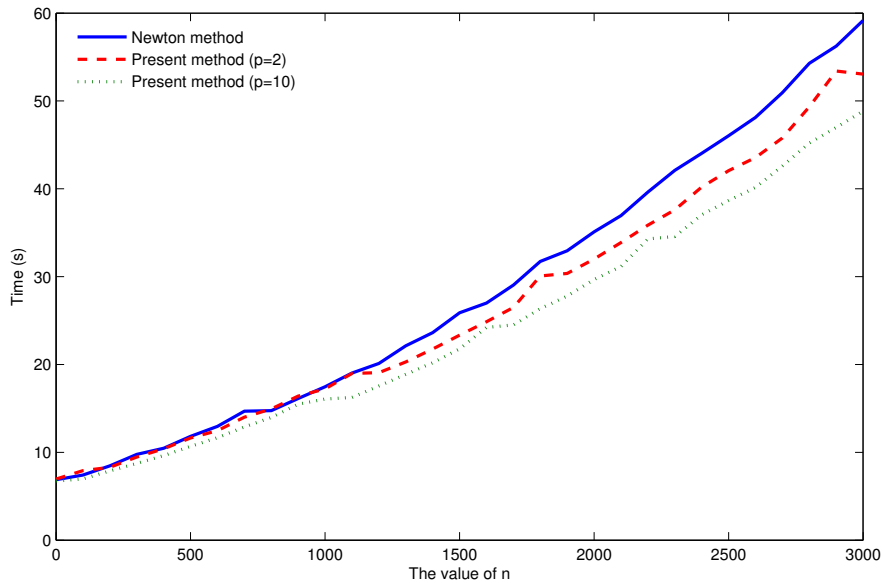


Figure 4.7: The results of comparisons of computational time, Example 4.4.

Figure 4.7 shows that as the number of columns become larger than the number of rows, the required computational times for current methods become smaller than that of Newton method.

Conclusion: A family of second-order iterative methods were developed based on Penrose equations (2.4a) and (2.4b) and written in terms of p -th root of matrix AX_k . Convergence properties were considered and numerical tests were made. Numerical results show that the number of iterations of current methods always less than that of Newton's method. Also, it is observed that the CPU time compared with Newton's method decreases when the number of columns is larger than the number of rows, this makes the current methods more efficient for such cases.

4.2 Second order iterative method

In this section, a new second order iterative method for computing the Moore-Penrose inverse is developed.

Let $A \in \mathbb{C}^{m \times n}$ and $X = A^\dagger \in \mathbb{C}^{n \times m}$. We use Equations (2.4a) and (2.4b) to obtain

$$X = XAX = X(AXA)X = X(AX)^2.$$

Hence, for arbitrary $\beta \in \mathbb{R}$ we have

$$X = X - \beta(X(AX)^2 - X).$$

From the last equation we get the following iterative method

$$X_{k+1} = X_k - \beta X_k((AX_k)^2 - I). \quad (4.11)$$

Assume the starting value of the iterative method (4.11) is

$$X_0 = \alpha A^*, \quad (4.12)$$

for an appropriate real number α .

Lemma 4.2.1. *The iterative scheme (4.11) with (4.12) satisfy the following relations*

$$XAX_k = X_k, \quad (4.13a)$$

$$X_kAX = X_k, \quad (4.13b)$$

$$(AX_k)^* = (AX_k), \quad (4.13c)$$

$$(X_kA)^* = (X_kA), \quad (4.13d)$$

where $k \geq 0$.

Proof. We use mathematical induction. For $k = 0$ we have $X_0 = \alpha A^*$ and all statements in (4.13) hold. Assume the statements are true for some integer k . Now we prove the statements for $k + 1$.

For (4.13a), we have

$$XAX_{k+1} = XA(X_k - \beta X_k((AX_k)^2 - I))$$

$$\begin{aligned}
 &= XAX_k - \beta XAX_k((AX_k)^2 - I) \\
 &= X_k - \beta X_k((AX_k)^2 - I) \\
 &= X_{k+1}.
 \end{aligned}$$

We prove (4.13b) in a similar way

$$\begin{aligned}
 X_{k+1}AX &= (X_k - \beta X_k((AX_k)^2 - I))AX \\
 &= X_kAX - \beta(X_k(AX_k)^2AX - X_kAX) \\
 &= X_k - \beta(X_k(AX_k)^2 - X_k) \\
 &= X_{k+1}.
 \end{aligned}$$

Now, for (4.13c)

$$\begin{aligned}
 (AX_{k+1})^* &= (A(X_k - \beta X_k((AX_k)^2 - I)))^* \\
 &= (AX_k)^* - \beta(((AX_k)^*)^2 - I)(AX_k)^* \\
 &= AX_k - \beta((AX_k)^2 - I)AX_k \\
 &= AX_k - \beta((AX_k)^3 - AX_k) \\
 &= AX_k - \beta AX_k((AX_k)^2 - I) \\
 &= AX_{k+1}.
 \end{aligned}$$

And (4.13d) can be verified in a similar way

$$\begin{aligned}
 (X_{k+1}A)^* &= ((X_k - \beta X_k((AX_k)^2 - I))A)^* \\
 &= (X_kA)^* - \beta A^*(((AX_k)^*)^2 - I)X_k^*
 \end{aligned}$$

$$\begin{aligned}
 &= X_k A - \beta(((X_k A)^*)^3 - A^* X_k^*) \\
 &= X_k A - \beta((X_k A)^3 - X_k A) \\
 &= X_{k+1} A.
 \end{aligned}$$

This completes the proof of the lemma. □

Now, we want to prove that the matrix sequence X_k defined by the iterative method (4.11) and the starting value (4.12) converges to the Moore-Penrose inverse $X = A^\dagger$.

Theorem 4.2.1. *Iterative method (4.11) with the starting value defined in (4.12) converges to the Moore-Penrose inverse $X = A^\dagger$ under the assumption*

$$\|(X - X_0)A\| < \frac{-3 + \sqrt{17}}{2}, \quad 0 < \beta \leq \frac{1}{2}. \quad (4.14)$$

For $\beta < \frac{1}{2}$ the method has a linear convergence, while for $\beta = \frac{1}{2}$ its convergence is quadratic.

Proof. Using Lemma 4.2.1 and substituting for X_k , we get

$$\|E_k\| = \|X - X_k\| = \|XAX - X_kAX\| \leq \|XA - X_kA\| \|X\| = \|t_k\| \|X\|,$$

where $t_k = XA - X_kA$.

Now using Lemma 4.2.1 and (4.11), we get

$$\begin{aligned}
 t_{k+1} &= XA - X_{k+1}A = XA - (X_k - \beta X_k((AX_k)^2 - I))A \\
 &= XA - X_kA + \beta X_k((AX_k)^2 - I)A \\
 &= t_k + \beta(X_kA)^3 - \beta(X_kA) \\
 &= t_k + \beta(XA - t_k)^3 - \beta(XA - t_k) \\
 &= t_k + \beta((XA)^3 - (XA)^2 t_k - XAt_kXA + XAt_k^2 \\
 &\quad - t_k(XA)^2 + t_kXAt_k + t_k^2XA - t_k^3) - \beta XA + \beta t_k
 \end{aligned}$$

$$\begin{aligned}
 &= t_k + \beta(XA - XAt_k - XAt_kXA + XAt_k^2 \\
 &\quad - t_kXA + t_kXAt_k + t_k^2XA - t_k^3) - \betaXA + \beta t_k \\
 &= t_k + \beta(-3t_k + 3t_k^2 - t_k^3) + \beta t_k \\
 &= (1 - 2\beta)t_k + 3\beta t_k^2 - \beta t_k^3. \tag{4.15}
 \end{aligned}$$

Let $s_k = \|t_k\|$, we require that $s_k \rightarrow 0$ as $k \rightarrow \infty$. Condition (4.14) implies $s_0 < \frac{-3 + \sqrt{17}}{2}$, then by mathematical induction we prove that $s_k < \frac{-3 + \sqrt{17}}{2}$.

From (4.15) and inductive method $s_k < \frac{-3 + \sqrt{17}}{2}$ we obtain

$$s_{k+1} \leq (1 - 2\beta)s_k + 3\beta s_k^2 + \beta s_k^3 < (1 - 2\beta)s_k + 2\beta s_k < s_k < \frac{-3 + \sqrt{17}}{2}. \tag{4.16}$$

Thus, s_k is a monotonically decreasing bounded sequence converging to s as $k \rightarrow \infty$ and $0 \leq s < \frac{-3 + \sqrt{17}}{2}$. From (4.16), we get

$$s \leq (1 - 2\beta)s + 3\beta s^2 + \beta s^3,$$

then

$$\beta s^3 + 3\beta s^2 - 2\beta s \geq 0.$$

This gives either $s = 0$ or $s \geq \frac{-3 + \sqrt{17}}{2}$. Thus, $s = 0$.

This complete the proof that $s_k \rightarrow 0$ as $k \rightarrow \infty$. Thus, $X_k \rightarrow X$ as $k \rightarrow \infty$.

From (4.15), we conclude that iterative method (4.11) has linear convergence if $\beta < \frac{1}{2}$, and its convergence is quadratic if $\beta = \frac{1}{2}$. \square

We need to write condition (4.14) in an equivalent form which does not contain the Moore-Penrose inverse X .

According to Lemma 2.5.1, necessary and sufficient condition for the convergence of

the iterative method (4.11) is $\rho((\alpha A^* - X)A) < \frac{-3 + \sqrt{17}}{2}$.

Theorem 4.2.2. *Let the eigenvalues of a matrix A^*A satisfy*

$$\sigma_1(A) \geq \dots \geq \sigma_r(A) > \sigma_{r+1}(A) = \dots = 0.$$

Condition $\rho((X - \alpha A^)A) < \frac{-3 + \sqrt{17}}{2}$ is satisfied under the assumptions*

$$\max_{1 \leq i \leq r} |1 - \alpha \lambda_i(A^*A)| < \frac{-3 + \sqrt{17}}{2}.$$

Proof. Let $P = XA$ and $Q = I - \alpha A^*A$. Since $P^2 = P$ and

$$\begin{aligned} PQ &= XA - \alpha XAA^*A = XA - \alpha(AXA)^*A \\ &= XA - \alpha A^*A \\ &= XA - \alpha A^*AXA \\ &= (I - \alpha A^*A)XA \\ &= QP, \end{aligned}$$

from Lemma 2.5.2 we can conclude that

$$\rho((X - \alpha A^*)A) \leq \rho(I - \alpha A^*A) = \max_{1 \leq i \leq r} |1 - \alpha \lambda_i(A^*A)| < \frac{-3 + \sqrt{17}}{2}.$$

□

In Theorem 4.2.1 we found that the condition of convergence restrict the usage of this method, since the condition

$$\|(X - X_0)A\| < \frac{-3 + \sqrt{17}}{2} \approx 0.56.$$

Next, we follow the idea of [5] to prove that the sequence X_k defined by the iterative method (4.11) and the starting value (4.12) still converges to the Moore-Penrose inverse

$X = A^\dagger$ under the condition

$$\|(X - X_0)A\| < 1.$$

Let us consider the following singular value decomposition of the matrix A of $\text{rank}(A) = r \leq \min\{m, n\}$

$$A = V \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} U^*, \quad S = \text{diag}(\sigma_1, \dots, \sigma_r), \quad \sigma_1 \geq \dots \geq \sigma_r > 0.$$

The Moore-Penrose is given by

$$A^\dagger = U \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^*.$$

Where U and V are unitary matrices. Using

$$X_0 = \alpha A^*,$$

in which α is a constant. We can deduce that each iterate of the method (4.11) has a singular value decomposition of the form

$$X_k = U S_k V^*, \quad S_k = \text{diag}(s_1^{(k)}, \dots, s_r^{(k)}),$$

where

$$S_0 = \alpha S,$$

and when $\beta = \frac{1}{2}$, we have

$$\begin{aligned} U S_{k+1} V^* &= U S_k V^* - \frac{1}{2} U S_k V^* ((V S S_k V^*)^2 - I) \\ &= \frac{3}{2} U S_k V^* - \frac{1}{2} U S_k S S_k S S_k V^*. \end{aligned}$$

Hence,

$$S_{k+1} = S_k \left(\frac{3}{2} I - \frac{1}{2} (S S_k)^2 \right).$$

Therefore, the diagonal matrices $R_k = S S_k = \text{diag}(r_1^{(k)}, \dots, r_r^{(k)})$ satisfy

$$R_{k+1} = g(R_k) = \frac{3}{2} R_k - \frac{1}{2} R_k^3,$$

that means

$$r_i^{(k+1)} = g(r_i^{(k)}) = \frac{3}{2}r_i^{(k)} - \frac{1}{2}r_i^{(k)3}. \quad (4.17)$$

Theorem 4.2.3. *For any initial point $r^{(0)} \in (0, \sqrt{\frac{5}{3}})$, the sequence $r^{(k+1)} = g(r^{(k)})$ is second order convergent to $r = 1$, in which the function $g(r)$ is defined by (4.17).*

Proof. The fixed points and the critical points of $g(r)$ are

$$g(r) = r \Rightarrow r = -1, 0, 1,$$

$$g'(r) = 0 \Rightarrow r = -1, 1.$$

We can find that 1 is local maximizer and 0 is local minimizer of $g(r)$. It is easy to see that the interval $(0, \sqrt{\frac{5}{3}})$ is mapped into itself.

Moreover, $g(r)$ is a continuous function on the interval $(0, \sqrt{\frac{5}{3}})$, and $|g'(r)| < 1$ on this interval, see Figure 4.8.

We conclude that the sequence $r^{(k+1)} = g(r^{(k)})$ is convergent to $r = 1$. On the other hand,

$$g(1) = 1, \quad g'(1) = 0,$$

implies that the convergence is second order by Lemma 2.5.4 .

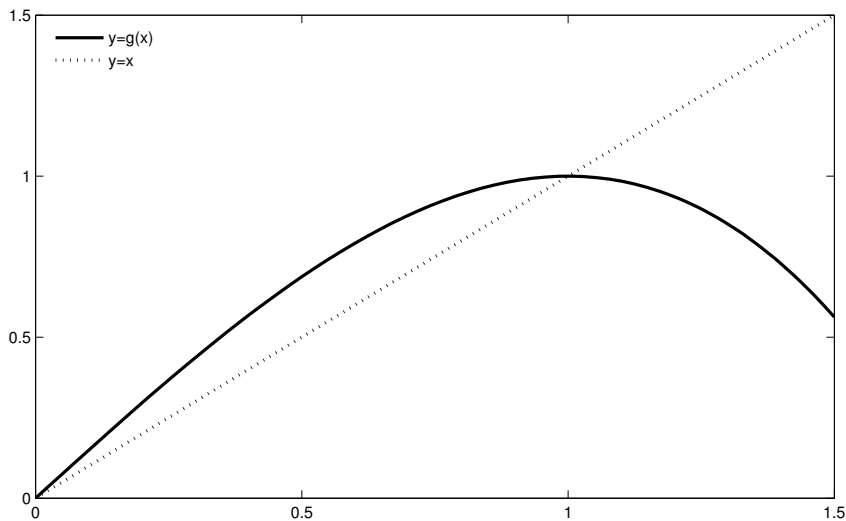


Figure 4.8: Graph of the function $y = g(x)$ and the line $y = x$.

□

Considering Theorem 4.2.3, we conclude that if $\alpha\sigma_1^2 = r_1^{(0)} \in (0, \sqrt{\frac{5}{3}})$, then $\alpha\sigma_i^2 = r_i^{(0)} \in (0, \sqrt{\frac{5}{3}})$, for all i , and

$$\lim_{k \rightarrow \infty} R_k = I.$$

Hence,

$$\lim_{k \rightarrow \infty} S_k = S^{-1},$$

so

$$\lim_{k \rightarrow \infty} X_k = U \begin{bmatrix} S^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^* = A^\dagger.$$

Hence, we have the following theorem.

Theorem 4.2.4. *Let A be an $m \times n$ nonzero complex matrix. If the initial approximation X_0 is defined by:*

$$X_0 = \alpha A^*, \quad \text{with } 0 < \alpha < \frac{\sqrt{\frac{5}{3}}}{\sigma_1^2}, \quad (4.18)$$

then

$$\|(X - X_0)A\| < 1,$$

and iterative method (4.11) converges to A^\dagger with second order when $\beta = \frac{1}{2}$, where σ_1^2 denotes the largest singular value of A .

Proof. Take $P = XA$ and $Q = I - \alpha A^*A$. Then $P^2 = P$ and $PQ = QP$, so we proved that

$$\rho((X - \alpha A^*)A) \leq \max_{1 \leq i \leq r} \|1 - \alpha \lambda_i(A^*A)\| = \max_{1 \leq i \leq r} \|1 - \alpha \sigma_i^2\|.$$

By using (4.18), we conclude that

$$\|(X - X_0)A\| \leq \rho((X - \alpha A^*)A) \leq \max_{1 \leq i \leq r} \|1 - \alpha \sigma_i^2\| < 1.$$

□

Example 4.5. Consider the matrix A of order (6×5) given by

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 \\ 1 & 3 & 4 & 6 & 2 \\ 2 & 3 & 4 & 5 & 3 \\ 3 & 4 & 5 & 6 & 4 \\ 4 & 5 & 6 & 7 & 6 \\ 6 & 6 & 7 & 7 & 8 \end{bmatrix},$$

of $\text{rank}(A) = 4$.

The choice $\alpha = 0.002$ satisfies the convergence criteria given by

$$\max_{1 \leq i \leq 4} |1 - \alpha \lambda_i(A^*A)| = 0.9999 < 1,$$

since the eigenvalues of the matrix A^*A are

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (640.6455, 17.0053, 0.3315, 0.0177).$$

The iterative method (4.11) when $\beta = \frac{1}{2}$ generates a sequence of iterates $\{X_k\}$ after 30 steps converging to the Moore-Penrose inverse A^\dagger given by

$$A^\dagger = \begin{bmatrix} 0.5 & -0.125 & -1 & 0.875 & -0.625 & 0.375 \\ -1 & 1.875 & -4.5 & 2.875 & -0.625 & 0.375 \\ 1.25 & -1.625 & 3.25 & -1.875 & 0.125 & -0.125 \\ -0.25 & 0.375 & -0.25 & 0.125 & 0.125 & -0.125 \\ -0.5 & -0.25 & 1.5 & -1.25 & 0.75 & -0.25 \end{bmatrix},$$

Example 4.6. Consider the ill-conditional Hilbert matrix A of order (5×5) appeared in Example 3.2

The iterative method (4.11) when $\beta = \frac{1}{2}$ generates a sequence of iterates $\{X_k\}$ after 68 steps converging to the Moore-Penrose inverse A^\dagger given by

$$A^\dagger = \begin{bmatrix} 25 & -300 & 1050 & -1400 & 630 \\ -300 & 4800 & -18900 & 26880 & -12600 \\ 1050 & -18900 & 79380 & -117600 & 56700 \\ -1400 & 26880 & -117600 & 179200 & -88200 \\ 630 & -12600 & 56700 & -88200 & 44100 \end{bmatrix}.$$

Example 4.7. Let $A = \text{rand}(50, 50)$, we have tested 50 times with MATLAB. The number of iterations and average of CPU time required for convergence are compared between (4.11) and (4.3) in figures 4.9 and 4.10, respectively.

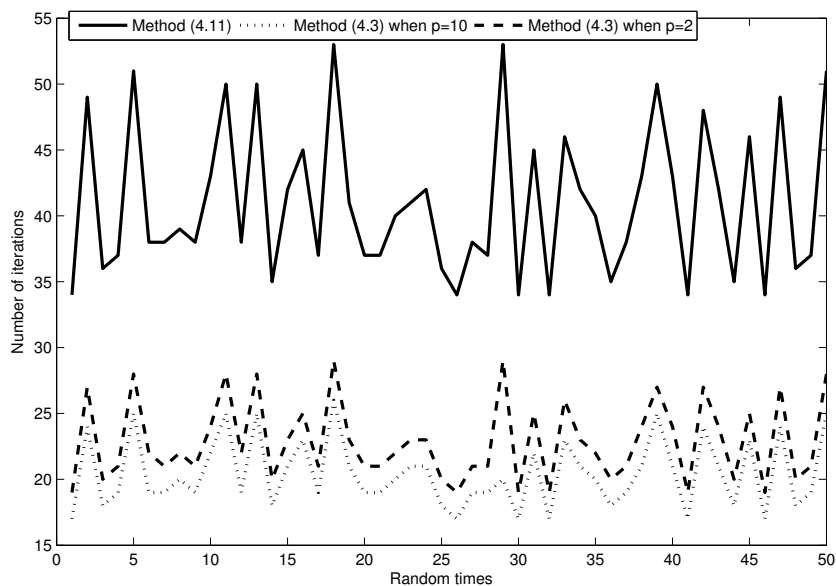


Figure 4.9: Comparison number of iteration, Example 4.7.

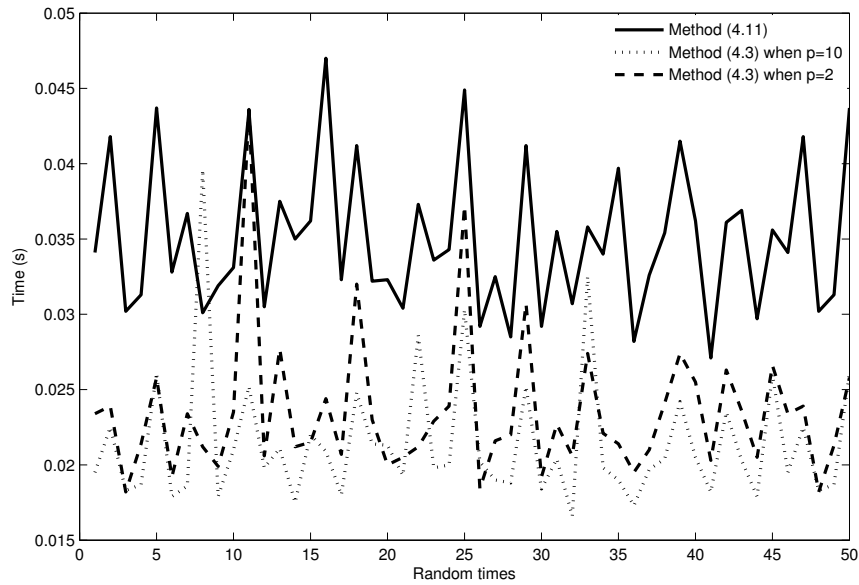


Figure 4.10: The results of comparisons of computational time, Example 4.7.

Conclusion: A second-order iterative methods were developed based on Penrose equations (2.4a) and (2.4b) and written in terms of square of matrix AX_k . Convergence properties were considered and numerical tests were made. It is observed that the family of second order methods (4.3) is more effective than the method (4.11).

Bibliography

- [1] A. Ben-Israeal, and T. Greville, *Generalized Inverses: Theory and Applications*, Springer-Verlag, NewYork, 2nd edition, (2003).
- [2] R. Burden, and J. Faires, *Numerical Analysis*, 9th edition. Brooks/Cole, Cengage Learning, Boston, (2011).
- [3] A. Cichocki, and B. Unbehauen, *Neural networks for optimization and signal processing*, New York: John Wiley and Sons, (1993).
- [4] G. Codevico, V. Pan, and M. Van Barel, *Newton-like iteration based on a cubic polynomial for structured matrices*, Numerical Algorithms, vol. 36, no. 4, 365-380, (2004).
- [5] H. Esmaeili, R. Erfanifar, and M. Rashidi, *A Fourth-order iterative method for computing the Moore-Penrose inverse*, Journal of Hyperstructures, (1), 52-67, (2017).
- [6] W. Gander, M. Gander, and F. Kwok, *Scientific Computing - An Introduction using Maple and MATLAB*, *Texts in Computational Science and Engineering 11*, Springer International Publishing Switzerland, (2014).
- [7] R. Kress, *Numerical Analysis*, Springer-Verlag, NewYork, (1998).
- [8] E. Krishnamurthy, and S. Sen, *Numerical Algorithms: Computations in Science and Engineering*, New Delhi, India: Affiliated East-West Press, (1986).
- [9] H. Li, T. Huang, Y. Zhang, X. Liu, and T. Gu, *Chebyshev-type methods and preconditioning techniques*, Applied Mathematics and Computation, (218), 260-270, (2011).
- [10] J. Marrero, R. Ben Taher, Y. El Khatabi, and M. Rachidi, *On explicit formulas of the principal matrix p th root by polynomial decompositions*, Applied Mathematics and Computation, (242), 435-443, (2014).

- [11] S. Miljkovic, M. Miladinovic, P. Stanimirovic, and I. Stojanovic, *Application of the pseudo-inverse computation in reconstruction of blurred images*, Filomat, (26), 453-465, (2012).
- [12] M. Petkovic, and P. Stanimirovic, *Two improvements of the iterative method for computing Moore–Penrose inverse based on Penrose equations*, Journal of Computational and Applied Mathematics, (267), 61-71, (2014).
- [13] M. Petkovic, and P. Stanimirovic, *Iterative method for computing the Moore-Penrose inverse based on Penrose equations*, Journal of Computational and Applied Mathematics, (235), 1604-1613, (2011).
- [14] V. Pan, and R. Schreiber, *An improved Newton iteration for the generalized inverse of a matrix, with applications*, SIAM Journal on Scientific and Statistical Computing, vol. 12, no. 5, 1109-1130, (1991).
- [15] V. Pan, F. Soleymani, and L. Zhao, *An efficient computation of generalized inverse of a matrix*, Applied Mathematics and Computation, (316), 89-101 (2018).
- [16] J. Rajagopalan, *An iterative algorithm for inversion of matrices*, [M.S.thesis]. Concordia University, Quebec, Canada, (1996).
- [17] I. Risteski, *A new generalized matrix inverse method for balancing chemical equations and thier stability*, Boletín de la Sociedad Chilena de Química, (2), 104-115, (2008).
- [18] P. Roland, and P. Beim, *Inverse problems in neural field theory*, SIAM Journal on Applied Dynamical Systems, (8), 1405-1433, (2009).
- [19] L. Sciavicco, and B. Siciliano, *Modelling and Control of robot manipulators*, London: Springer-Verlag, (2000).
- [20] S. Sen, H. Agarwal, and S. Sen, *Chemical equation balancing: An integer programming approach*, Mathematical and Computer Modelling, (44), 678-691, (2006).
- [21] G. Shultz, *iterative Berechmmg der reziproken matrix*, Journal of Applied Mathematics and Mechanics, (13), 57-59, (1933).
- [22] M. Sharifi, M. Arab, and F. Khaksar Haghani, *Finding generalized inverses by a fast and efficient numerical method*, Journal of Computational and Applied Mathematics, (279), 187-191, (2014).

- [23] F. Soleymani, M. Sharifi, and S. Shateyi, *Approximating the inverse of a square matrix with application in computation of the Moore-Penrose Inverse*, Journal of Applied Mathematics, 2-8 (2014).
- [24] F. Soleimani, P. Stanimirovic, and F. Soleymani , *Some matrix iterations for computing generalized inverses and balancing chemical equations*, Algorithms, (8), 982-998 (2015).
- [25] F. Soleymani, P. Stanimirovic, and M. Z. Ullah , *An accelerated iterative method for computing weighted Moore-Penrose inverse*, Applied Mathematics and Computation, (222), 365-371 (2013).
- [26] S. Srivastava, and D. Gupta, *Higher order iterations for Moore-Penrose inverses* , Journal of Applied Mathematics Informatics, (32), 171-184, (2014).
- [27] P. Stanimirovic, and D. Cvetkovic-Ilic, *Successive matrix squaring algorithm for computing outer inverses*, Applied Mathematics and Computation, (203), 19-29, (2008).
- [28] F. Toutounian, and F. Soleymani, *An iterative method for computing the approximate inverse of a square matrix and the Moore-Penrose inverse of a non-Square matrix*, Applied Mathematics and Computation, (224), 671-680, (2013).