Algorithmic and Computational Questions Concerning Matrices

by

Levan Bezhanishvili

Submitted to

Central European University

Department of Mathematics And Its Applications

In partial fulfilment of the requirements for the degree of
Master of Science

Supervisor: Professor Pal Hegedus

Budapest, Hungary

2012

Contents

Chapter 1

Introduction

The aim of the following thesis will be investigation the Moore-Penrose (generalized) inverses of matrices by different iteration processes.

1.1 The necessity of iteration processes for a matrix inverse

Except usual ways of calculating a matrix inverse, there are methods which compute it iteratively. A reasonable question would be why these iterative methods would be needed, especially when having in mind that by using them we cannot get an exact answer, only an approximation. In reply to this question, one should consider the role that computers play in mathematics nowadays. Using computers for the computation of different numerical problems started in 1950s and 1960s. Nowadays it is hard to imagine that one can solve complex numerical problems without using some kind of computer language programs. Without doubt, computers had their enormous contribution to the field of mathematics. Yet, using them in mathematics is not without problems. Generally, as it turned out, computer programs cannot give an exact numerical value of division of two numbers if it has long fractional part. In that case a programming language always rounds it. For example $1000000/999999 = 1.000001000001\ldots$ In this case most of computer programs would give as an output: 1. But why is it so important to compute problems exactly, when in most cases we do not need the exact numerical results. For example: For a mathematical problem instead of using value 1.999999999999 we can round it and take number 2. In many cases this would not have a substantial influence on the solution. In some situations,

however, it can considerably damage the final outcome. Below is presented an example of this situation.
The following is from [3].
Consider the system:

$$\begin{cases} 0.001x + y = 1 \\ x + y = 2 \end{cases}$$

We see that the solution is $x = 1000/999 \approx 1,\ y = 998/999 \approx 1$ which does not change much if the coefficients are altered slightly.
Let us solve this system of equations by usual row reduction algorithm. Adding a multiple of the first to the second row gives the system on the left below, then dividing by 999 and rounding to 3 places on $998/999 = 0.99899 \approx 1.00$ gives the system on the right:

$$\begin{cases} 0.001x + y = 1 \\ -999y = -998 \end{cases} \quad \begin{cases} 0.001x + y = 1 \\ y = 1.00 \end{cases}$$

The solution for the last system is $x = 0, y = 1$ which is wildly inaccurate. But as we know this row reduction algorithm is quite usable for solving systems of linear equations by different computer programs. Having this in mind, a reasonable question would be: Why does a problem of correctness appear in some equations? On what does it depend? How can we avoid it? This paper will respond to these questions.
The main problem is ill-conditioning. When a matrix is ill-conditioned, a modern computer may not be able to work with it. The main indicator of measuring ill-conditioning is the condition number. If the condition number is not too big, then we can say that a matrix is well-conditioned.

Definition 1.1.1. *Condition number for square matrix A is*

$$K(A) = \|A\|_{op} \cdot \|A^{-1}\|_{op}.$$

In our presented example the condition number for the initial matrix $A = \begin{pmatrix} 0.001 & 1 \\ 1 & 1 \end{pmatrix}$ is equal $K(A) \approx 4$. After using row reduction algorithm the initial matrix A is transformed into a new matrix $B = \begin{pmatrix} 0.001 & 1 \\ 0 & 1 \end{pmatrix}$, note that the condition number has increased significantly $K(B) \approx 2002$. Thus, we can conclude that the computer property of rounding long fractional parts of numbers can change the solution significantly. Therefore, in order to avoid these troubles, there exist iterative methods of computing inverses of matrices. In my thesis, I will discuss several of them.

1.2 Iterative methods for the Moore-Penrose pseudoinverse

For a matrix beside the usual matrix inverse there exist other inverses. In this thesis I will discuss the Moore-Penrose pseudoinverse. I took this paragraph from [6] verbatim. [The great advantage of this inverse is that every matrix has one, square or not, full rank or not. The idea of a generalized inverse of a singular matrix goes back to E.H. Moore in a paper published in 1920. He investigated the idea of a "general reciprocal" of a matrix again in a paper in 1935. Independently, R. Penrose rediscovered Moore's idea in 1955].
A main use of the MoorePenrose inverse is to compute a least squares solution(see definition below) for a system of linear equations that lacks a unique solution. It is also used to find the minimum (Euclidean) norm solution(see definition below) to a system of linear equations with multiple solutions.

Definition 1.2.1. *(minimum norm)[6] We say x_0 is a minimum norm solution of $Ax = b$ iff x_0 is a solution and $\|x_0\| \leq \|x\|$ for all solutions x of $Ax = b$.*

Definition 1.2.2. *(least square)[6] A vector x_0 is called a least squares solution of the system of linear equations $Ax = b$ iff $\|Ax_0 - b\| \leq \|Ax - b\|$ for all vectors x.*

I took this paragraph from [6] verbatim. When $Ax = b$ has a solution, A^+b is the solution of minimum norm. When $Ax = b$ does not have a solution, A^+b gives the least squares solution of minimum norm. (by A^+ we denote A matrix's pseudoinverse)
There are several methods of computing the Moore-Penrose inverse. The mostly used are: "Singular Value Decomposition(SVD)" and "QR Factorization". During the process of computing SVD and QR factorization methods for getting the Moore-Penrose inverse of a matrix, we may get a matrix which will be ill-conditioned and after this a computer might give as an output a result which will be significantly different from the actual result. Therefore, scientists started investigating iteration processes in order to avoid this kind of problems. The pioneers of exploring iterative methods of pseudoinverse were Adi Ben-Israel and Dan Cohen. They published their first article about it in 1966 and nowadays many people are working with it. There are many papers and articles connected to iteration processes of the Moore-Penrose inverse.

In this thesis one can see different iterative methods for computing the pseudoinverse, charts and numerical results of them for different parameters, as well as exploring each process by searching parameters, and comparison of different iterative methods.

Chapter 2

Iterative methods for a regular matrix inverse

The following section is based on [7].

2.1 Gauss-Seidel iterative method

Consider the system of n linear equations:

$$\sum_{q=1}^{n} a_{pq}x_q = b_q, \qquad 1 \leq p \leq n$$

The Gauss-Seidel method is:

$$x_p^{r+1} = -\frac{1}{a_{pp}}[\sum_{q=1}^{p-1} a_{pq}x_q^{r+1} + \sum_{q=p+1}^{n} a_{pq}x_q^{r} - b_p].$$

It is known that Gauss-Seidel iterative method is convergent if and only if the matrix $A_{n\times n} = [a_{pq}]$ is symmetric and positive-definite.
Let us now consider the system of linear equations

$$CX = D.$$

Suppose C is regular, but not positive definite, so the Gauss-Seidel method is not convergent. Let us multiply the equation by C^T from the left. We get

$$AX = B,$$

where $A = C^TC$ and $B = C^TD$. Notice that A is both symmetric and positive-definite. Truly:

Symmetricity: $A^T = (C^T C)^T = C^T C^{TT} = C^T C = A$.
Positive definiteness: $y^T A y = y^T C^T C y = (Cy)^T C y > 0$ for non zero y.
Therefore we can apply Gauss-Seidel iterative method for the equation $AX = B$, which is convergent and the result yields the solution of the equation $CX = D$.
We can use this technique for finding C^{-1}.
It is the solution of $CX = I$, so the solution of $AX = C^T$, where $A = C^T C$.
We solve the equation $AX = C^T$ column by column using the Gauss-Seidel iterative method.

2.2 Newton's iterative method

Let $A \in \mathbb{R}_{n\times n}$. If A is invertible then with suitable starting matrix X_0 the following iteration process is convergent to A^{-1}.

$$X_{n+1} = X_n(2I - AX_n).$$

In fact it is convergent if the eigenvalues of $I - AX_0$ are at most 1. According to Victor Pan and Robert Schreiber [4], a good starting matrix is

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

where

$$\|A\|_1 = \max_i (\sum_j |a_{ij}|),$$

$$\|A\|_\infty = \max_j (\sum_i |a_{ij}|).$$

Chapter 3

The Moore-Penrose pseudoinverse

3.1 Description of the Moore-Penrose pseudoinverse

Definition 3.1.1. *(The Moore-Penrose inverse)[6]*
Let A be any matrix in $\mathbb{C}^{m\times n}$. We say A has **The Moore-Penrose inverse** *(or just* **Pseudoinverse** *for short) if and only if there is a matrix A^+ in $\mathbb{C}^{m\times n}$ such that*

a) $AA^+A = A$
b) $A^+AA^+ = A^+$
c) $(AA^+)^* = AA^+$
d) $(A^+A)^* = A^+A.$

Theorem 3.1.2. *[6]For any matrix $A \in \mathbb{C}^{m\times n}$ there always exists the Moore-Penrose inverse which is unique.*

(Proof is not discussed here.)

Theorem 3.1.3. *[6]If a matrix is invertible then inverse of this matrix is equal to its pseudoinverse.*

(Proof is not discussed here.)

Theorem 3.1.4. *[6]Let $A \in \mathbb{C}^{m\times n}$. Then*

1. $(AA^+)^2 = AA^+ = (AA^+)^*$.
2. $(I_m - AA^+)^2 = A^+A = (A^+A)^*$.
3. $(A^+A)^2 = A^+A = (A^+A)^*$.
4. $(I_n - A^+A)^2 = (I_n - A^+A) = (I_n - A^+A)^*$.
5. $A^{++} = A$.
6. $(A^*)^+ = (A^+)^*$.
7. $(A^*A)^+ = A^+A^{*+}$.
8. $A^* = A^*AA^+ = A^+AA^*$.
9. $A^+ = (A^*A)^+A^* = A^*(AA^*)^+$.

3.2 Singular Value Decomposition(SVD)

Singular Value Decomposition method is factorization of matrices in the complex space.

Definition 3.2.1. *[6]A square matrix $X \in \mathbb{C}_{n\times n}$ is unitary if*

$$X^*X = XX^* = I_{n\times n},$$

where X^* is conjugate transpose of X and $I_{n\times n}$ is identity matrix of dimension n.

Remark: In other words, U is a unitary matrix iff $U^* = U^{-1}$.

Theorem 3.2.2. *[8]Let A be $m \times n$ matrix $(m \geq n)$ of rank r. Then A can be represented in the following form:*

$$A = U\Sigma V^* = U \left[\begin{array}{cccc|c} \delta_1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \delta_r & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] V^*. \tag{3.1}$$

Where U is a $m \times m$ unitary matrix, $\Sigma = diag(\delta_1, \delta_2, \dots, \delta_r)$ $\delta_1 > 0, \delta_2 > 0 \dots \delta_r > 0$ is a $n \times n$ rectangular diagonal matrix with nonnegative elements, V is a $n \times n$ unitary matrix.

(The proof of the following theorem is beyond the scope of this thesis.)
Define:

$$X = V\Sigma^+U^* = V \left[\begin{array}{ccc|c} \frac{1}{\delta_1} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \frac{1}{\delta_r} & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] U^*. \tag{3.2}$$

Theorem 3.2.3. *[8]X is Moore-Penrose inverse of the matrix A,* $A^+ = X$.

Proof. We have to check the four conditions of the Moore-Penrose inverse.

1) $AXA = A$.
$AXA = U\Sigma V^*V\Sigma^+U^*U\Sigma V^* = U\Sigma I_{n\times n}\Sigma^+ I_{m\times m}\Sigma V^* = U\Sigma\Sigma^+\Sigma V^*$
$= U\Sigma V^*$.

2) $XAX = X$.
$XAX = V\Sigma^+U^*U\Sigma V^*V\Sigma^+U^* = V\Sigma^+ I_{m\times m}\Sigma I_{n\times n}\Sigma^+U^*$
$= V\Sigma^+\Sigma\Sigma^+U^* = V\Sigma^+U^* = X$.

3) $(XA)^* = XA$.
$(XA)^* = (A)^*(X)^* = (U\Sigma V^*)^*(V\Sigma^+U^*)^* = (V\Sigma^*U^*)(U(\Sigma^+)^*V^*)$
$= (V\Sigma U^*)(U\Sigma^+V^*) = XA$.

4) $(AX)^* = AX$.
$(AX)^* = (X)^*(A)^* = (V\Sigma^+U^*)^*(U\Sigma V^*)^* = (U(\Sigma^+)^*V^*)(V\Sigma^*U^*)$
$= (U\Sigma^+V^*)(V\Sigma U^*) = AX$. □

3.3 QR factorization

Theorem 3.3.1. *(QR factorization)[6] Let $A \in C^{m\times n}$, with $n \le m$. Then there is a matrix $Q \in C^{m\times n}$ with orthonormal columns and an upper triangular matrix R in $C^{n\times n}$ such that $A = QR$. Moreover, if $n = m$, Q is square and unitary. Even more, if A is square and nonsingular, R may be selected so as to have positive real numbers on the diagonal. In this case, the factorization is unique.*

(The proof of the following theorem is beyond the scope of this thesis.)

Lemma 3.3.2. *[6]If A is of full column rank then A^*A is regular.*

Theorem 3.3.3. *[6]*$A^+ = (A^*A)^{-1}A^*$.

Proof. Let us check the conditions of the pseudoinverse.

1) $AA^+A = A(A^*A)^{-1}(A^*A) = A.$

2) $A^+AA^+ = (A^*A)^{-1}(A^*A)(A^*A)^{-1}A^* = (A^*A)^{-1}A^* = A^+.$

3) $(A^+A)^* = ((A^*A)^{-1}A^*A)^* = (I_{n\times n})^* = I_{n\times n} = (A^*A)^{-1}(A^*A)$
$= ((A^*A)^{-1}A^*)A) = A^+A.$

4) $(AA^+)^* = (A(A^*A)^{-1}A^*)^* = (A^*)^*((A^*A)^{-1})^*A^* = A((A^*A)^*)^{-1}A^*$
$= A(A^*A)^{-1}A^* = AA^+.$ □

Consider the case when A is full rank. Then by the theorem of factorization A can be factorized of the following form $A = QR$ where Q and R are unitary and upper triangular matrices. $A^* = R^*Q^* \Rightarrow A^*A = R^*Q^*QR = R^*R$. If we substitute the last result into the formula of A^+, we will get:

$$A^+ = (A^*A)^{-1}A^* = (R^*R)^{-1}A^*.$$

When A is not full rank then A can be factorized of the following form

$$A = [Q_1, Q_2]\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1R_1.$$

Where R is upper triangular and $Q \in \mathbb{R}^{m\times m}$ is orthogonal. Here $Q_1 \in \mathbb{R}^{m\times n}$. The decomposition $A = Q_1R_1$ is called the *economy* QR factorization. For *economy* QR factorization the pseudoinverse has the following form

$$A^+ = R_1^{-1}Q_1^*.$$

Chapter 4

Iterative methods for the Moore-Penrose matrix inverse

The following section is based on [5].

4.1 Iterative method for computing the Moore-Penrose inverse based on Penrose equations

Assume $A \in C^{m\times n}$ and $X = A^{+} \in C^{m\times n}$, as we know

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

For arbitrary $\beta \in \mathbb{R}$ it holds

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

Equivalently

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

We can formulate the following iteration process

$$X_{k+1} = (I - \beta X_k A)^* X_k + \beta X_k. \tag{4.1}$$

Let us assume that the starting value for the iterative method (4.1) is

$$X_0 = \beta A^*. \tag{4.2}$$

Lemma 4.1.1. *For the sequence generated by (4.1) and (4.2) the following holds.*

$$X_k A = (X_k A)^*, \quad XAX_k = X_k, \quad X_k AX = X_k, \quad k \geq 0. \tag{4.3}$$

Proof. In order to prove the equations, we use induction.
For $k = 0$ the statement is true: $X_0 A = \beta A^* A = (\beta A^* A)^* = (X_0 A)^*$. Let us assume $X_k A = (X_k A)^*$ for k. Then

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

We proved the first statement of Lemma. Similarly we prove the second one.

For $k = 0$, $\quad XAX_0 = \beta XAA^* = \beta A^* = X_0$.
Assume the second statement of Lemma holds for k. Let us prove it for $k+1$:

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

Similarly we prove the third statement of the lemma.

For $k = 0$, $\quad X_0 AX = \beta A^* AX = \beta A^* = X_0$.
Assume it holds for k. For $k + 1$:

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

□

By using Lemma 4.1.1 equation (4.1) can be rewritten in the following form:

$$X_{k+1} = (I - \beta X_k A) X_k + \beta X_k = (1 + \beta) X_k - \beta X_k A X_k. \tag{4.4}$$

Theorem 4.1.2. *Iterative method (4.4) with the starting value (4.2) converges to the Moore-Penrose inverse $X = A^+$ under the assumptions*

$$\|(X_0 - X)A\| < 1, \qquad 0 < \beta \leqslant 1. \tag{4.5}$$

For $\beta < 1$ the method has a linear convergence, when $\beta = 1$ its convergence is quadratic. The first and the second order terms, corresponding to the error estimation of (4.4) are equal to:

$$error_1 = (1 - \beta) E_k, \qquad error_2 = -\beta E_k A E_k. \tag{4.6}$$

Proof. For the first part of the theorem it is sufficient to show that $\|(X_0 - X)A\| \to 0$ when $n \to +\infty$. Using (4.3) and the properties of the pseudoinverse, we obtain

$$\|X_{k+1} - X\| = \|X_{k+1}AX - XAX\| \leq \|X_{k+1}A - XA\|\|X\|. \tag{4.7}$$

Using (4.3) and (4.4) we obtain

$$\begin{aligned} X_{k+1}A - XA &= (1+\beta)X_kA - \beta X_kAX_kA - XA \\ &= -(\beta X_kA - XA)(X_kA - XA). \end{aligned}$$

Taking into account

$$\beta X_kA - XA = \beta(X_kA - XA) - (1-\beta)XA,$$

and using (4.3) we have

$$X_{k+1}A - XA = -\beta(X_kA - XA)^2 + (1-\beta)(X_kA - XA).$$

Let us define $E_k = X_k - X$, then the last statement can be written in the following form

$$E_{k+1}A = -\beta(E_kA)^2 + (1-\beta)E_kA. \tag{4.8}$$

Let $t_k = \|E_kA\|$. Our aim is to show that $t_k \to 0$ when $k \to +\infty$. First we show that $t_k < 1$. We will show it by using induction. From (4.5) we get $t_0 = \|(X_0 - X)A\| < 1$. Now let us assume that $t_k < 1$. From (4.8) we obtain

$$t_{k+1} \leq \beta(t_k)^2 + (1-\beta)t_k < \beta t_k + (1-\beta)t_k = t_k. \tag{4.9}$$

We have proved $t_{k+1} \leq t_k < 1$. Therefore t_k is a decreasing sequence, which is bounded $t_k \geq 0$. Therefore we know that this sequence is convergent, $t_k \to t$, when $k \to +\infty$, for some $0 \leq t < 1$. If we take limit on both sides of (4.9), we get

$$t \leq \beta t^2 + (1-\beta)t \quad \Longrightarrow \quad 0 \leq t(t-1).$$

Hence $t \geq 1$ or $t = 0$, and therefore we conclude $t = 0$. It completes the proof of $t_k \to 0$.
By (4.7), $\|X_k - X\| \leq t_k\|X\|$, so we conclude $X_k \to X$. We proved the first part of the theorem.
Substituting $X + E_k = X_k$ in (4.4), we obtain the following expression for the error matrix E_k:

$$E_{k+1} = (1+\beta)E_k - \beta XAE_k - \beta E_kAX - \beta E_kAE_k,$$

which implies

$$error_1 = (1+\beta)E_k - \beta XAE_k - \beta E_k AX.$$

$$error_2 = -\beta E_k A E_k.$$

Using Lemma 4.1.1 and $E_k = X_k - X$ we get

$$\begin{aligned} error_1 &= (1+\beta)(X_k - X) - \beta XA(X_k - X) - \beta(X_k - X)AX \\ &= (1-\beta)(X_k - X) = (1-\beta)E_k. \end{aligned}$$

All claims of the theorem are justified. □

Lemma 4.1.3. *Let $\varepsilon > 0$ and $M \in C^{n\times n}$ be given. There exist minimum one matrix norm $\|\cdot\|$ such that*

$$\rho(M) \leq \|M\| \leq \rho(M) + \varepsilon.$$

Where

$$\rho(M) = max(|\lambda_1(M)|, |\lambda_2(M)|, \ldots, |\lambda_n(M)|).$$

Lemma 4.1.4. *If matrices $S, P \in \mathbb{C}^{n\times n}$ are such that $PS = SP, \quad P = P^2$ then*

$$\rho(PS) \leq \rho(S).$$

Lemma 4.1.5. *Let us assume that the eigenvalues of A^*A satisfy*

$$lambda_1(A^*A) \geq \cdots \geq lambda_r(A^*A) = \cdots = lambda_n(A^*A) = 0.$$

Then $\rho((\beta A^* - X)A) < 1$ is satisfied under the assumption

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

Let us denote $s_k = \|E_k\|$ and $d_k = \|E_{k+1} - E_k\|$.

Theorem 4.1.6. *Iterative method (4.4) with starting value (4.2) satisfies*

$$\lim_{k\to\infty} \frac{t_{k+1}}{t_k} = \lim_{k\to\infty} \frac{s_{k+1}}{s_k} = \lim_{k\to\infty} \frac{d_{k+1}}{d_k} = 1 - \beta.$$

Proof. From (4.8)

$$E_{k+1}A = -\beta(E_k A)^2 + (1/\beta)E_k A,$$

we can conclude

$$\begin{aligned} t_{k+1} = \|E_{k+1}A\| &\geq \|(1-\beta)E_k A\| - \|\beta(E_k A)^2\| \\ &\geq (1-\beta)\|E_k A\| - \beta\|E_k A\|^2 = t_k(1 - \beta - \beta t_k). \end{aligned}$$

On the other hand

$$t_{k+1} = \|E_{k+1}A\| \leq \|(1-\beta)E_kA\| + \|\beta(E_kA)^2\|$$
$$\leq (1-\beta)\|E_kA\| + \beta\|E_kA\|^2 = t_k(1-\beta+\beta t_k).$$

From these two inequalities we imply

$$1-\beta-\beta t_k \leq \frac{t_{k+1}}{t_k} \leq 1-\beta+\beta t_k.$$

From Theorem (4.1.2) we know that $t_k = \|E_kA\| \to 0$. If we take limit of the previous equation we get that $t_{k+1}/t_k \to 1-\beta$, when $k \to \infty$.
By using Theorem (4.1.2) we can write

$$E_{k+1} = (1-\beta)E_k - \beta E_kAE_k.$$

The previous equation implies

$$1-\beta-\beta t_k\frac{\|E_kAE_k\|}{\|E_k\|} \leq \frac{\|E_{k+1}\|}{\|E_k\|} \leq 1-\beta+\beta\frac{\|E_kAE_k\|}{\|E_k\|} \tag{4.10}$$

From $\|E_kAE_k\| \leq \|E_k\|^2\|A\|$ and $\|E_k\| \to 0$, it follows

$$0 \leq \lim_{k\to\infty}\frac{\|E_kAE_k\|}{\|E_k\|} \leq \lim_{k\to\infty}\|E_k\|\|A\| = 0.$$

If we take limit on both sides of (4.10) and use the previous equation, then we conclude $s_{k+1}/s_k \to 1-\beta$, when $k \to \infty$.
In order to verify the third statement about the sequence d_k, we are starting by using $X_{k+1} - X_k = E_{k+1} - E_k$ and (4.6) which together imply
$d_k = \|E_{k+1} - E_k\| = \|(1-\beta)E_k - \beta E_kAE_k - E_k\| = \beta\|E_k + E_kAE_k\|$.
Analogously to the above, we get

$$\lim_{k\to\infty}\frac{d_k}{s_k} = \lim_{k\to\infty}\frac{d_k}{\|E_k\|} = \beta.$$

We obtain

$$\lim_{k\to\infty}\frac{d_{k+1}/s_{k+1}}{d_k/s_k} = 1,$$

this implies

$$\lim_{k\to\infty}\frac{d_{k+1}}{d_k} = \lim_{k\to\infty}\frac{d_{k+1}/s_{k+1}}{d_k/s_k}\cdot\frac{s_{k+1}}{s_k} = 1-\beta.$$

□

The following section is based on [1].

4.2 On Iterative Computation of Generalized Inverses and Associated Projections

Let $A \in C^{m\times n}$ be a nonzero matrix. Let r denote the rank of A. Let $\lambda_1(A^*A) \geq \lambda_2(A^*A) \geq \ldots \geq \lambda_n(A^*A)$ be the eigenvalues of A^*A. As it is known $\lambda_r(A^*A) > 0$ and $\lambda_i(A^*A) = 0, \quad i = r+1, \ldots n$. In this section we will use the Euclidean matrix norm $\|A\| = \sqrt{\lambda_1(A^*A)}$.

Theorem 4.2.1. *Let $\alpha \in \mathbf{R}$ satisfy*

$$0 < \alpha < \frac{2}{\lambda_1(A^*A)}, \tag{4.11}$$

then the sequence

$$X_k = \alpha \sum_{p=0}^{k} A^*(1 - \alpha AA^*)^p, \qquad k = 0, 1, \ldots \tag{4.12}$$

converges to A^+ when $k \to \infty$.

(Proof is a consequence of theorems 4.2.3, 4.2.4 and 4.2.5.)

Theorem 4.2.2. *Let us assume condition (4.11) Then the sequence*

$$Y_0 = \alpha A^*, \tag{4.13}$$

$$Y_{k+1} = Y_k(2I - AY_k), \qquad k = 0, 1, \ldots. \tag{4.14}$$

converges to A^+ when $k \to \infty$.

(Proof is a consequence of theorem 4.2.5.)

Theorem 4.2.3. *(a) The process (4.12) is of the first order*

$$\|A^+ - X_{k+1}\| < \|A^+ - X_k\|. \tag{4.15}$$

(b) The process (4.14) is of the second order

$$\|A^+ - Y_{k+1}\| < \|A\|\|A^+ - Y_k\|^2. \tag{4.16}$$

Proof. (a)

$$\begin{aligned} X_{k+1} &= \alpha \sum_{p=0}^{k+1} A^*(1-\alpha AA^*)^p \\ &= \alpha A^* + \alpha \sum_{p=1}^{k+1} A^*(I-\alpha AA^*)^p \\ &= \alpha A^* + (\alpha \sum_{p=0}^{k} A^*(I-\alpha AA^*)^p)(I-\alpha AA^*) \\ &= \alpha A^* + X_k(I-\alpha AA^*). \end{aligned} \tag{4.17}$$

From the last statement consider with $A^+AA^* = A^*$, we get

$$\begin{aligned} A^+ - X_{k+1} &= A^+ - \alpha A^* - X_k(I-\alpha AA^*) \\ &= (A^+ - X_k)(I-\alpha AA^*), \quad k = 1, 2 \ldots \end{aligned} \tag{4.18}$$

which, because of condition (4.11) proves (4.15).

Now, let us prove the second part of the theorem.
Notice, that $Y_k = Y_kAA^+ = A^+AY_k$, because $Y_k = C_kA^*$ for some matrix C_k and at the same time $Y_k = A^*B_k$ for some matrix B_k. $\quad k = 0, 1, \ldots$
Therefore the following statement holds:

$$\begin{aligned} A^+ - Y_{k+1} &= A^+AA^+ - Y_k - Y_k + Y_kAY_k \\ &= A^+AA^+ - A^+AY_k - Y_kAA^+ + Y_kAY_k \\ &= (A^+ - Y_k)A(A^+ - Y_k). \end{aligned} \tag{4.19}$$

If we take norm on both sides of the previous equation we get (b). $\square$

Theorem 4.2.4. *The following statement holds*

$$Y_k = X_{2^k-1}. \tag{4.20}$$

(Proof is derived by using induction on k.)
Remark. By using Euler's Identity

$$(1+x)\prod_{p=1}^{k-1}(1+x^{2^p}) = \sum_{p=0}^{2^k-1} x^p, \quad |x| < 1 \tag{4.21}$$

and 4.20, we obtain:

$$Y_k = \alpha A^*[I + (I - \alpha AA^*)] \prod_{p=1}^{k-1} [I + (I - \alpha AA^*)^{2^p}], \tag{4.22}$$

which converges to A^+.

Theorem 4.2.5. *The following for $k = 0, 1, \dots$ is true:*

$|A^+ - Y_k\| \leq \frac{\sqrt{\lambda_1(A^*A)}(1-\alpha\lambda_r(A^*A))^{2^k}}{\lambda_r(A^*A)} \qquad for \qquad 0 < \alpha \leq \frac{2}{\lambda_1(A^*A)+\lambda_r(A^*A)}.$

$|A^+ - Y_k\| \leq \frac{\alpha\sqrt{\lambda_1(A^*A)}(1-\alpha\lambda_1(A^*A))^{2^k}}{2-\alpha\lambda_1(A^*A)} \qquad for \qquad \frac{2}{\lambda_1(A^*A)+\lambda_r(A^*A)} < \alpha < \frac{2}{\lambda_1(A^*A)}.$

Proof. From Theorems (4.2.1) and (4.2.3) it follows

$$A^+ - Y_k = \alpha \sum_{p=2^k}^{\infty} A^*(I - \alpha AA^*)^p = \alpha \sum_{p=2^k}^{\infty} A^*(AA^+ - \alpha AA^*)^p. \ \ k = 0, 1, \dots. \tag{4.23}$$

The following statements holds:

$$\|AA^+ - \alpha AA^*\| = |1 - \alpha\lambda_r(A^*A)| \qquad 0 < \alpha \leq \frac{2}{\lambda_1(A^*A) + \lambda_r(A^*A)},$$

$$\|AA^+ - \alpha AA^*\| = |1 - \alpha\lambda_1(A^*A)| \qquad \frac{2}{\lambda_1(A^*A) + \lambda_r(A^*A)} < \alpha < \frac{2}{\lambda_1(A^*A)}.$$

Truly, let $W = Im(AA^*)$, and $B = AA^+ - \alpha AA^*$. Hence $B^* = AA^+ - \alpha AA^*$. It is known that AA^+ is the orthogonal projection onto W.

$$B^*Bx = \lambda x \quad \Rightarrow \quad x \in W \quad \Rightarrow \quad AA^+x = x.$$

So

$$(AA^+ - \alpha AA^*)^2 x = \lambda x,$$
$$x - 2\alpha AA^*x + \alpha^2 AA^*AA^*x = \lambda x,$$
$$(1 - \alpha AA^*)^2 x = \lambda x,$$
$$\lambda = (1 - \alpha\lambda_i(A^*A))^2,$$
$$\|B\| = \sqrt{\lambda_1(B^*B)} = max|1 - \alpha\lambda_i(A^*A)|.$$

Finally we obtain

$$\begin{cases} \|AA^+ - \alpha AA^*\| = |1 - \alpha\lambda_r(A^*A)| & 0 < \alpha \leq \frac{2}{\lambda_1(A^*A)+\lambda_r(A^*A)} \\ \|AA^+ - \alpha AA^*\| = |1 - \alpha\lambda_1(A^*A)| & \frac{2}{\lambda_1(A^*A)+\lambda_r(A^*A)} < \alpha < \frac{2}{\lambda_1(A^*A)} \end{cases}$$

If we take norm on both sides of the (4.23) and use the previous equation, we obtain

$$\begin{aligned}\|A^+ - Y^k\| &\leq \alpha\|A^*\| \sum_{p=2^k}^{\infty} \|AA^+ - \alpha AA^*\|^p \\ &\leq \frac{\alpha\|A^*\|\|AA^+ - \alpha AA^*\|^{2^k}}{1 - \|AA^+ - \alpha AA^*\|}\end{aligned} \tag{4.24}$$

$$\leq \begin{cases} \frac{\sqrt{\lambda_1(A^*A)}(1-\alpha\lambda_r(A^*A))^{2^k}}{\lambda_r(A^*A)} & When \quad 0 < \alpha \leq \frac{2}{\lambda_1(A^*A)+\lambda_r(A^*A)} \\ \frac{\alpha\sqrt{\lambda_1(A^*A)}(1-\alpha\lambda_1(A^*A))^{2^k}}{2-\alpha\lambda_1(A^*A)} & When \quad \frac{2}{\lambda_1(A^*A)+\lambda_r(A^*A)} < \alpha < \frac{2}{\lambda_1(A^*A)} \end{cases} \quad k = 0, 1, \ldots$$

□

Definition 4.2.6. *α is optimal if it minimizes $\|AA^+ - \alpha AA^*\|$.*

Theorem 4.2.7. *The optimal α is*

$$\alpha_0 = \frac{2}{\lambda_1(A^*A) + \lambda_r(A^*A)}, \tag{4.25}$$

for which

$$\|A^+ - Y^k\| \leq \frac{\sqrt{\lambda_1(A^*A)}}{\lambda_r(A^*A)}(1 - \alpha_0\lambda_r(A^*A))^{2^k}. \tag{4.26}$$

The following section is based on [2].

4.3 Iterative Methods for Computing Generalized Inverses related to Optimization Methods

Let $\mathcal{A}$ be a complex C^* Algebra with unit.

Definition 4.3.1. *An $a \in \mathcal{A}$ is called relatively regular if there exists $b \in \mathcal{A}$ such that $aba = a$. Such b is called an inner generalized inverse of a.*

It is known that a is relatively regular if and only if it has Moore-Penrose inverse.

Lemma 4.3.2. *Let $a \in \mathcal{A}$ be relatively regular. The following holds:*
(1) $(a^*)^+ = (a^+)^*$;
(2) $(a^*a)^+ = a^+(a^*)^+$;
(3) $a^+ = (a^*a)^+a^*$;
(4) $(a^*a)^+$ commutes with every element of $\mathcal{A}$ which commutes with a^*a;
(5) a^*a is invertible in the algebra $\beta = a^+a\mathring{A}a^+a$ and $(a^*a)^+ = (a^*a)_\beta^{-1}$;
(6) $a^* = a^+aa^* = a^*aa^+$;

Definition 4.3.3. *If $d \in \mathcal{A}$ is self-adjoint then upper and lower bounds of the spectrum of d in the algebra $\mathcal{A}$ are*

$$M_{\mathcal{A}}(d) = sup\{< x, dx >: \|x\| = 1\}.$$

$$m_{\mathcal{A}}(d) = inf\{< x, dx >: \|x\| = 1\}.$$

Theorem 4.3.4. *Suppose that $a \in \mathcal{A}$ is relatively regular, $x_0, c \in \mathcal{A}$ are arbitrary, $\mathcal{B} = a^+a\mathcal{A}a^+a$ and $(\lambda_n)_n$ is a sequence of positive numbers such that*

$$0 < \epsilon \leq \lambda_n \leq 2\min\{[M_{\mathcal{B}}(a^*a)]^{-1}, [m_{\mathcal{B}}(a^*a)]^{-1}\} - \delta,$$

holds for $\forall$ n and some $\epsilon, \delta > 0$. Then the iterative method

$$x_{n+1} = x_n - \lambda_n a^*(ax_n - c), \tag{4.27}$$

converges to $a^+c + (1 - a^+a)x_0$, consequently, $\lim\limits_{n\to\infty} x_n = a^+c$ if and only if $a^+ax_0 = x_0$.

Proof. Assume first that $a^+ax_0 = x_0$. We claim that $a^+ax_n = x_n \quad (\forall n)$. Indeed, we prove it by induction on n.
Assume $a^+ax_n = x_n$, then

$$\begin{aligned} a^+ax_{n+1} &= a^+a(x_n - \lambda_n a^*(ax_n - c)) \\ &= a^+ax_n - \lambda_n a^+aa^*(ax_n - c) \\ &= x_n - \lambda_n a^*(ax_n - c)) = x_{n+1}, \end{aligned}$$

as claimed. We compute

$$a^*ax_{n+1} - a^*c = a^*ax_n - a^*c - \lambda_n a^*a(a^*ax_n - a^*c) = (1 - \lambda_n a^*a)(a^*ax_n - a^*c).$$

If we multiply the previous statement by $(a^*a)^+$ from the left side and use Lemma 4.3.2(3)-(4), we obtain:

$$(a^*a)^+a^*ax_{n+1} - (a^*a)^+a^*c = (a^*a)^+(1 - \lambda_n a^*a)(a^*ax_n - a^*c),$$

$$a^+ax_{n+1} - a^+c = (1 - \lambda_n a^*a)(a^*a)^+(a^*ax_n - a^*c),$$
$$x_{n+1} - a^+c = (1 - \lambda_n a^*a)(a^+ax_n - a^+c),$$

and:

$$x_{n+1} - a^+c = a^+a(x_{n+1} - a^+c) = a^+a(1 - \lambda_n a^*a)a^+a(x_n - a^+c),$$
$$\Downarrow$$
$$\|x_{n+1} - a^+c\| \leq \|a^+a(1 - \lambda_n a^*a)a^+a\|\|x_n - a^+c\|.$$

The convergence will be implied once we find a universal upper bound $\|a^+a(1-\lambda_n a^*a)a^+a\| \leq q < 1$. We know that

$$\|a^+a(1-\lambda_n a^*a)a^+a\| = max\{|M_{\mathcal{B}}[a^+a(1-\lambda_n a^*a)a^+a]|, |m_{\mathcal{B}}[a^+a(1-\lambda_n a^*a)a^+a]|\}.$$

Notice that

$$M_{\mathcal{B}}[a^+a(1 - \lambda_n a^*a)a^+a] = 1 - \lambda_n m_{\mathcal{B}}(a^*a),$$

and

$$m_{\mathcal{B}}[a^+a(1 - \lambda_n a^*a)a^+a] = 1 - \lambda_n M_{\mathcal{B}}(a^*a).$$

Hence, the existence of the bound $\|a^+a(1 - \lambda_n a^*a)a^+a\| \leq q < 1$ is a consequence of the assumed bounds for λ_n :

$$0 < \epsilon \leq \lambda_n \leq 2\min\{[M_{\mathcal{B}}(a^*a)]^{-1}, [m_{\mathcal{B}}(a^*a)]^{-1}\} - \delta.$$

We have proved the first part of the theorem.
Now, Let us assume that $a^+ax_0 \neq x_0$. We denote $x' = a^+ax$ and $x'' = (1 - a^+a)x$ for any $x \in \mathcal{A}$. Now we have $x_0'' \neq 0$. Let us prove by induction that $x_{n+1}'' = x_0''$. First we need to show that $x_1'' = x_0''$.

$$\begin{aligned} x_1'' &= (1 - a^+a)x_1 = (1 - a^+a)(x_0 - \lambda_0 a^*(ax_0 - c)) \\ &= (1 - a^+a)x_0 - \lambda_0(1 - a^+a)a^*(ax_0 - c) \\ &= x_0'' - \lambda_0(a^* - a^+aa^*)(ax_0 - c) \\ &= x_0'' - \lambda_0(a^* - a^*)(ax_0 - c) = x_0'', \end{aligned}$$

assume that $x_n'' = x_0''$. We need to show $x_{n+1}'' = x_0''$.
Truly,

$$\begin{aligned} x_{n+1}'' &= (1 - a^+a)x_{n+1} = (1 - a^+a)(x_n - \lambda_n a^*(ax_n - c)) \\ &= (1 - a^+a)x_n - \lambda_n(1 - a^+a)a^*(ax_n - c) \\ &= x_n'' - \lambda_n(a^* - a^*)(ax_n - c) = x_n'' = x_0''. \end{aligned}$$

Hence

$$x_{n+1} = x^{'}_{n+1} + x^{''}_{n+1} = x^{'}_{n+1} + x^{''}_{0}.$$

Also

$$a^+ax^{'}_0 = a^+aa^+ax_0 = a^+ax_0 = x^{'}_0.$$

We compute:

$$\begin{aligned} x^{'}_{n+1} = a^+ax_{n+1} &= a^+a(x_n - \lambda_n a^*(ax_n - c)) \\ &= a^+ax_n - \lambda_n a^+aa^*(ax_n - c) \\ = x^{'}_n - \lambda_n a^*(aa^+ax_n - c) &= x^{'}_n - \lambda_n a^*(ax^{'}_n - c). \end{aligned}$$

Therefore we have $a^+ax^{'}_0 = x^{'}_0$ and $x^{'}_{n+1} = x^{'}_n - \lambda_n a^*(ax^{'}_n - c)$.
Now for the sequence $x^{'}_n$ we can apply the first part of the current theorem. We get

$$\lim_{n\to\infty} x^{'}_n = a^+c.$$

Therefore

$$\lim_{n\to\infty} x_{n+1} = \lim_{n\to\infty} x^{'}_{n+1} + \lim_{n\to\infty} x^{''}_0 = a^+c + (1 - a^+a)x_0.$$

□

Theorem 4.3.5. *Suppose that $a \in \mathcal{A}$ is relatively regular, $x_0, c \in \mathcal{A}$ are arbitrary, and $(\lambda_n)_n$ is a bounded sequence of positive numbers. Then the iterative method*

$$x_{n+1} = x_n - (\lambda_n + a^*a)^{-1}a^*(ax_n - c), \tag{4.28}$$

converges to $a^+c + (1 - a^+a)x_0$, consequently, $\lim_{n\to\infty} x_n = a^+c$ if and only if $a^+ax_0 = x_0$.

Proof. Let $a^+ax_0 = x_0$. By induction on n we prove $a^+ax_n = x_n \quad \forall n$. Assume $a^+ax_0 = x_0$ and $a^+ax_n = x_n$, we need to show that $a^+ax_{n+1} = x_{n+1}$.

$$\begin{aligned} a^+ax_{n+1} &= a^+ax_n - a^+a(\lambda_n + a^*a)^{-1}a^*(ax_n - c) \\ &= x_n - a^+a(\lambda_n + a^*a)^{-1}a^*(ax_n - c). \end{aligned}$$

We need to show that

$$a^+a(\lambda_n + a^*a)^{-1}a^*(ax_n - c) = (\lambda_n + a^*a)^{-1}a^*(ax_n - c).$$

Truly,

$$\begin{aligned}
a^*(ax_n - c) &= a^+aa^*(ax_n - c) \\
&= a^+a(\lambda_n + a^*a)(\lambda_n + a^*a)^{-1}a^*(ax_n - c) \\
&= (\lambda_n a^+a + a^+aa^*a)(\lambda_n + a^*a)^{-1}a^*(ax_n - c) \\
&= (\lambda_n a^+a + a^*aa^+a)(\lambda_n + a^*a)^{-1}a^*(ax_n - c) \\
&= (\lambda_n + a^*a)a^+a(\lambda_n + a^*a)^{-1}a^*(ax_n - c).
\end{aligned}$$

We obtained:

$$a^*(ax_n - c) = (\lambda_n + a^*a)a^+a(\lambda_n + a^*a)^{-1}a^*(ax_n - c).$$

Finally, we have:

$$(\lambda_n + a^*a)^{-1}a^*(ax_n - c) = a^+a(\lambda_n + a^*a)^{-1}a^*(ax_n - c).$$

We have proved the statement: $a^+ax_n = x_n, \quad \forall\ n.$

Now we compute:

$$\begin{aligned}
a^*ax_{n+1} - a^*c &= a^*ax_n - a^*c - a^*a(\lambda_n + a^*a)^{-1}(a^*ax_n - a^*c) \\
&= \lambda_n(\lambda_n + a^*a)^{-1}(a^*ax_n - a^*c) \\
&= (1 - a^*a(\lambda_n + a^*a)^{-1}))(a^*ax_n - a^*c).
\end{aligned} \tag{4.29}$$

We know:

$$\begin{aligned}
(\lambda_n + a^*a)(\lambda_n + a^*a)^{-1} &= 1, \\
\lambda_n(\lambda_n + a^*a)^{-1} + a^*a(\lambda_n + a^*a)^{-1} &= 1, \\
\lambda_n(\lambda_n + a^*a)^{-1} &= 1 - a^*a(\lambda_n + a^*a)^{-1}.
\end{aligned}$$

If we substitute last result into (4.29) we obtain

$$a^*ax_{n+1} - a^*c = \lambda_n(\lambda_n + a^*a)^{-1}(a^*ax_n - a^*c).$$

If we multiply the previous equation by $(a^*a)^+$ from the left side and use Lemma 4.3.2(3) we obtain:

$$(a^*a)^+a^*ax_{n+1} - (a^*a)^+a^*c = (a^*a)^+\lambda_n(\lambda_n + a^*a)^{-1}(a^*ax_n - a^*c),$$

$$\begin{aligned}
a^+ax_{n+1} - a^+c &= a^*a^+\lambda_n(\lambda_n + a^*a)^{-1}(a^*ax_n - a^*c), \\
x_{n+1} - a^+c &= a^*a^+\lambda_n(\lambda_n + a^*a)^{-1}(a^*ax_n - a^*c).
\end{aligned}$$

Because of Lemma 4.3.2(4) if a^*a commutes with $(\lambda_n + a^*a)^{-1}$ then $(a^*a)^+$ will commute with the same element. Truly:

$(a^*a)^+$ commutes with $(\lambda_n + a^*a)^{-1}$. Because:

$$a^*a = a^*a,$$
$$a^*a(\lambda_n + a^*a)(\lambda_n + a^*a)^{-1} = a^*a,$$
$$(\lambda_n a^*a + a^*aa^*a)(\lambda_n + a^*a)^{-1} = (\lambda_n + a^*a)(\lambda_n + a^*a)^{-1}a^*a,$$
$$(\lambda_n + a^*a)a^*a(\lambda_n + a^*a)^{-1} = (\lambda_n + a^*a)(\lambda_n + a^*a)^{-1}a^*a,$$

If we multiply the previous equation by $(\lambda_n + a^*a)^{-1}$, we obtain:

$$a^*a(\lambda_n + a^*a)^{-1} = (\lambda_n + a^*a)^{-1}a^*a.$$

We have:

$$\begin{aligned} x_{n+1} - a^+c &= \lambda_n(\lambda_n + a^*a)^{-1}(a^*a)^+(a^*ax_n - a^*c) \\ &= \lambda_n(\lambda_n + a^*a)^{-1}(a^+ax_n - a^+c) \\ &= \lambda_n(\lambda_n + a^*a)^{-1}(ax_n - a^+c). \end{aligned}$$

We got:

$$x_{n+1} - a^+c = \lambda_n(\lambda_n + a^*a)^{-1}(ax_n - a^+c).$$

Now we have:

$$x_{n+1} - a^+c = a^+a(x_{n+1} - a^+c)$$
$$= \lambda_n a^+a(\lambda_n + a^*a)^{-1}a^+a(ax_n - a^+c).$$

Let $\mathcal{B} = a^+a\mathcal{A}a^+a$. Because a^+a is invertible in $\mathcal{B}$ (Lemma 4.3.2(5)), we know that $m_{\mathcal{B}}(a^*a) > 0$. The following holds

$$\begin{aligned} \|x_{n+1} - a^+c\| &\leq \lambda_n M_\beta[a^+a(\lambda_n + a^*a)^{-1}a^+a]\|x_n - a^+c\| \\ &= \frac{\lambda_n}{\lambda_n + m_\beta(a^*a)}\|x_n - a^+c\|. \end{aligned}$$

Because the sequence λ_n in bounded and the function $t \longrightarrow \frac{t}{t+m_\beta(a^*a)}$ is increasing, therefore we conclude that there exists $q \in \mathbb{R} \quad 0 < q < 1$, such that

$$\|x_{n+1} - a^+c\| \leq q\|x_n - a^+c\|.$$

Therefore $\lim\limits_{n\to\infty} x_n = a^+c$.
The first part of the theorem is proved.

Now, suppose that $a^+ax_0 \neq x_0$. We define $x^{'} = a^+ax$ and $x^{''} = (1-a^+a)x$ for any $x \in \mathcal{A}$. Now we have $x_0^{''} \neq 0$. We conclude that:

$$\begin{aligned} x_1^{'} = a^+ax_1 &= a^+a(x_0 - (\lambda_0 + a^*a)^{-1}a^*(ax_0 - c)) \\ &= a^+ax_0 - a^+a(\lambda_0 + a^*a)^{-1}a^*(aa^+ax_0 - c) \\ &= x_0^{'} - a^+a(\lambda_0 + a^*a)^{-1}a^*(x_0^{'} - c) \\ &= x_0^{'} - (\lambda_0 + a^*a)^{-1}a^*(x_0^{'} - c). \end{aligned}$$

As the correctness of the last equation, we only need to show that

$$a^+a(\lambda_0 + a^*a)^{-1} = (\lambda_0 + a^*a)^{-1}a^+a.$$

Truly:

$$\begin{aligned} a^+a(\lambda_0 + a^*a)(\lambda_0 + a^*a)^{-1} &= a^+a, \\ (\lambda_0 + a^*a)a^+a(\lambda_0 + a^*a)^{-1} &= a^+a, \\ a^+a(\lambda_0 + a^*a)^{-1} &= (\lambda_0 + a^*a)^{-1}a^+a. \end{aligned}$$

Therefore: $x_1^{'} = x_0^{'} - (\lambda_0 + a^*a)^{-1}a^*(x_0^{'} - c)$ and $x_1 = x_1^{'} + x_0^{''}$. By induction on n we obtain $x_n = x_n^{'} + x_0^{''} \quad \forall\ n$. Now for the sequence $x_n^{'}$ we can apply the first part of the current theorem. We get

$$\lim_{n\to\infty} x_n^{'} = a^+c.$$

Therefore

$$\lim_{n\to\infty} x_n = \lim_{n\to\infty} x_n^{'} + \lim_{n\to\infty} x_0^{''} = a^+c + (1-a^+a)x_0.$$

$\square$

Lemma 4.3.6. *If* 0 *is not a point of accumulation of* $\sigma_{\mathcal{A}}(a^*a)$*, then*

$$\lim_{\lambda\to 0}(\lambda + a^*a)^{-1}a^* = \lim_{\lambda\to 0} a^*(\lambda + aa^*)^{-1} = a^+.$$

Theorem 4.3.7. *Let* $a \in \mathcal{A}$ *be relatively regular, let* $(\alpha_n)_n$ *be strongly decreasing to 0 and let* $(\beta_n)_n$ *be a bounded sequence of positive numbers such that* $\beta_n - \alpha_n > 0 \quad \forall\ n$*. Consider the iterative method*

$$x_{n+1} = x_n - (\beta_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n x_n). \tag{4.30}$$

There are two possible cases.
(a) if $a^+ax_0 = x_0$*, then* $\lim\limits_{n\to\infty} x_n = a^+$.
(b) if $a^+ax_0 \neq x_0$*, then* $\lim\limits_{n\to\infty} x_n = a^+ + e^{\sum_{n=0}^{\infty} ln(1-\frac{\alpha_n}{\beta_n})}(1-a^+a)x_0$*. In this case* $\lim\limits_{n\to\infty} x_n = a^+$ *if and only if* $\sum\limits_{n=0}^{\infty} \frac{\alpha_n}{\beta_n}$ *is divergent.*

Proof. Let $y_n = (\alpha_n + a^*a)^{-1}a^*$. From Lemma 4.3.6 we get that $\lim_{n\to\infty} y_n = a^+$. Notice that

$$\begin{aligned} x_{n+1} &= x_n - (\beta_n + a^*a)^{-1}(\alpha_n + a^*a)[x_n - (\alpha_n + a^*a)^{-1}a^*] \\ &= x_n - (\beta_n + a^*a)^{-1}(\alpha_n + a^*a)(x_n - y_n). \end{aligned}$$

Now we compute

$$\begin{aligned} x_{n+1} - y_n &= x_n - y_n - (\beta_n + a^*a)^{-1}(\alpha_n + a^*a)(x_n - y_n) \\ &= (\beta_n + a^*a)^{-1}[\beta_n + a^*a - (\alpha_n + a^*a)](x_n - y_n). \end{aligned}$$

Consequently, we get

$$x_{n+1} - y_n = (\frac{\beta_n}{\beta_n - \alpha_n} + \frac{a^*a}{\beta_n - \alpha_n})^{-1}(x_n - y_n).$$

(a) Suppose that $a^+ax_0 = x_0$. Notice that $a^+ay_n = y_n$ holds for all n. By induction on n we get that $a^+ax_n = x_n$ is satisfied for all n. Truly:

$$\begin{aligned} a^+ax_{n+1} &= a^+ax_n - (\beta_n + a^*a)^{-1}a^+a(a^*ax_n - a^* + \alpha_n x_n) \\ &= x_n - (\beta_n + a^*a)^{-1}(a^+aa^*ax_n - a^+aa^* + \alpha_n a^+ax_n) \\ &= x_n - (\beta_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n x_n) = x_{n+1}. \end{aligned}$$

Hence we have

$$x_{n+1} - y_n = a^+a(x_{n+1} - y_n) = a^+a(\frac{\beta_n}{\beta_n - \alpha_n} + \frac{a^*a}{\beta_n - \alpha_n})^{-1}a^+a(x_n - y_n).$$

Again, let $\mathcal{B} = a^+a\mathcal{A}a^+a$; where a^*a is invertible in $\mathcal{B}$ and $m_{\mathcal{B}}(a^*a) > 0$. Notice that

$$\begin{aligned} \|a^+a(\frac{\beta_n}{\beta_n - \alpha_n} + \frac{a^*a}{\beta_n - \alpha_n})^{-1}a^+a\| &= M_{\mathcal{B}}(a^+a(\frac{\beta_n}{\beta_n - \alpha_n} + \frac{a^*a}{\beta_n - \alpha_n})^{-1}a^+a) \\ &= (\frac{\beta_n}{\beta_n - \alpha_n} + \frac{m_{\mathcal{B}}(a^*a)}{\beta_n - \alpha_n})^{-1} = \frac{\beta_n - \alpha_n}{\beta_n + m_{\mathcal{B}}(a^*a)} \leq \frac{\beta_n}{\beta_n + m_{\mathcal{B}}(a^*a)}. \end{aligned}$$

Because the sequence β_n is bounded and the function $t \longrightarrow \frac{t}{t + m_{\mathcal{B}}(a^*a)}$ is increasing, therefore we conclude that there exist $q \in \mathbb{R}\quad 0 < q < 1$, such that

$$\|a^+a(\frac{\beta_n}{\beta_n - \alpha_n} + \frac{a^*a}{\beta_n - \alpha_n})^{-1}a^+a\| \leq q < 1,$$

is satisfied for all n. For an arbitrary $\epsilon > 0$, there exists some n_0 such that $\|y_n - a^+\| < \epsilon$ holds for all $n \geq n_0$. This implies

$$\begin{aligned}
\|x_{n+1} - a^+\| &\le \|x_{n+1} - y_n\| + \|y_n - a^+\| \\
&\le q\|x_n - y_n\| + \|y_n - a^+\| \\
&\le q\|x_n - a^+\| + (1+q)\|y_n - a^+\| \\
&\le q\|x_n - a^+\| + (1+q)\epsilon \\
&\le q^{n-n_0+1}\|x_{n_0} - a^+\| + \epsilon(1+q)/(1-q).
\end{aligned}$$

We see that $\lim\limits_{n\to\infty} x_n = a^+$.

(b) Now, lets assume that $a^+ax_0 \neq x_0$. We denote by $x^{'}$ and $x^{''}$, $x^{'} = a^+ax$ and $x^{''} = (1 - a^+a)x$ for any $x \in \mathcal{A}$. Now we compute

$$x^{'}_{n+1} = a^+ax_{n+1} = a^+a(x_n - (\beta_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n x_n))$$

$$= a^+ax_n - a^+a(\beta_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n x_n)$$

$$= x^{'}_n - (\beta_n + a^*a)^{-1}a^+a(a^*ax_n - a^* + \alpha_n x_n)$$

$$= x^{'}_n - (\beta_n + a^*a)^{-1}(a^+aa^*ax_n - a^+aa^* + \alpha_n a^+ax_n)$$

$$= x^{'}_n - (\beta_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n a^+ax_n)$$

$$= x^{'}_n - (\beta_n + a^*a)^{-1}(a^*aa^+ax_n - (\alpha_n + a^*a)(\alpha_n + a^*a)^{-1}a^* + \alpha_n a^+ax_n)$$

$$= x^{'}_n - (\beta_n + a^*a)^{-1}((a^*a + \alpha_n)a^+ax_n - (\alpha_n + a^*a)(\alpha_n + a^*a)^{-1}a^*)$$

$$= x^{'}_n - (\beta_n + a^*a)^{-1}(a^*a + \alpha_n)(a^+ax_n - (\alpha_n + a^*a)^{-1}a^*)$$

$$= x^{'}_n - (\beta_n + a^*a)^{-1}(a^*a + \alpha_n)(x^{'}_n - y_n).$$

Notice that $a^+ax^{'}_0 = x^{'}_0$. Therefore if we apply the first part of the theorem we get $\lim\limits_{n\to\infty} x^{'}_n = a^+$. We compute

$$x^{''}_{n+1} = (1 - a^+a)x_{n+1} = (1 - a^+a)(x_n - (\beta_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n x_n)$$

$$= (1 - a^+a)x_n - (\beta_n + a^*a)^{-1}(1 - a^+a)(a^*ax_n - a^* + \alpha_n x_n)$$

$$= (1 - a^+a)x_n - (\beta_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n x_n - a^+aa^*ax_n + a^+aa^* - \alpha_n a^+ax_n)$$

$$= (1 - a^+a)x_n - (\beta_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n x_n - a^*ax_n + a^* - \alpha_n a^+ax_n)$$

$$= (1 - a^+a)x_n - (\beta_n + a^*a)^{-1}(\alpha_n x_n - \alpha_n a^+ax_n)$$

$$= (1 - a^+a)x_n - \alpha_n(\beta_n + a^*a)^{-1}(1 - a^+a)x_n$$

$$= x^{''}_n - \alpha_n(\beta_n + a^*a)^{-1}x^{''}_n$$

$$= (I - \alpha_n(\beta_n + a^*a)^{-1})x^{''}_n.$$

Notice that

$$(I-\alpha_n(\beta_n+a^*a)^{-1})x_n^{''}=(\beta_n-\alpha_n)(\beta_n+a^*a)^{-1}x_n^{''}.$$

Truly:

$$(a^*a-a^*aa^+a)x_n=0,$$

$$(\beta_n-\alpha_n-\beta_n-a^*a+\alpha_n)(I-a^+a)x_n=0,$$

$$[(\beta_n-\alpha_n)(\beta_n+a^*a)^{-1}-(\beta_n+a^*a)^{-1}(\beta_n+a^*a)+\alpha_n(\beta_n+a^*a)^{-1}]x_n^{''}=0,$$

$$[(\beta_n-\alpha_n)(\beta_n+a^*a)^{-1}-I+\alpha_n(\beta_n+a^*a)^{-1}]x_n^{''}=0,$$

$$(\beta_n-\alpha_n)(\beta_n+a^*a)^{-1}x_n^{''}=(I-\alpha_n(\beta_n+a^*a)^{-1})x_n^{''}.$$

Finally, we obtained:

$$x_{n+1}^{''}=(\beta_n-\alpha_n)(\beta_n+a^*a)^{-1}x_n^{''}.$$

From

$$\begin{aligned}\beta_n+a^*a&=(\beta_n+a^*a)a^+a+(\beta_n+a^*a)(1-a^+a)\\&=(\beta_n+a^*a)a^+a+\beta_n(1-a^+a),\end{aligned}$$

it is easy to verify that $(\beta_n+a^*a)^{-1}=[(\beta_n+a^*a)a^+a]_\beta^{-1}+(1-a^+a)/\beta_n$, where $[(\beta_n+a^*a)a^+a]_\beta^{-1}$ is an ordinary inverse of $(\beta_n+a^*a)a^+a$ in algebra $\mathcal{B}$. We obtain

$$\begin{aligned}x_{n+1}^{''}&=(\beta_n-\alpha_n)([(\beta_n+a^*a)a^+a]_\beta^{-1}+(1-a^+a)/\beta_n)x_n^{''}\\&=\frac{(\beta_n-\alpha_n)}{\beta_n}(\beta_n[(\beta_n+a^*a)a^+a]_\beta^{-1}+I-a^+a)x_n^{''}\\&=\frac{(\beta_n-\alpha_n)}{\beta_n}x_n^{''}=(\prod_{k=0}^{n}(1-\frac{\alpha_k}{\beta_k}))x_n^{''}.\end{aligned}$$

□

Chapter 5

Investigation of Iterative Methods for the Moore-Penrose Pseudoinverse, Numerical Results and Charts

In this chapter we will show that all iterative methods for the Moore-Penrose inverse discussed in this present work are based on Newton's method for the matrix inversion. In addition, numerical results from the iterative processes will be presented. Numerical examples are derived taking the following stopping criterion $\|X_{k+1} - X_k\| < \varepsilon$, where X_k and X_{k+1} are two successive results for an iterative process and ε is some small fixed real number.
The following matrix will be used through out the chapter.

$$A = \begin{pmatrix} 3 & 1 & 4 & 9 \\ 1 & 2 & 3 & 4 \\ 0 & -2 & -2 & 0 \\ -1 & 0 & -1 & -4 \end{pmatrix} \tag{5.1}$$

Hence

$$A^{+} = \begin{pmatrix} \frac{8}{9} & -\frac{47}{54} & -\frac{7}{27} & \frac{61}{54} \\ -\frac{4}{9} & \frac{14}{27} & -\frac{1}{27} & -\frac{13}{27} \\ \frac{4}{9} & -\frac{19}{54} & -\frac{8}{27} & \frac{35}{54} \\ -\frac{1}{3} & \frac{7}{18} & \frac{2}{9} & -\frac{11}{18} \end{pmatrix} \tag{5.2}$$

5.1 Investigation of the Iterative Method (4.14)

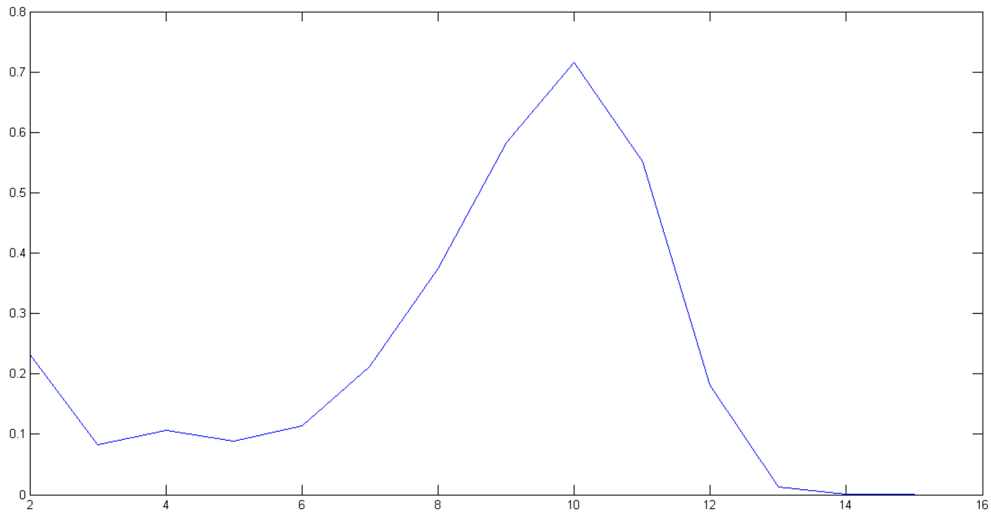
We can observe that Newton's iterative method for nonsingular matrix inversion described in section 2.2 is the same as Ben-Israel's and Cohen's method

(4.14) described in the section 4.2.

$$Y_{k+1} = Y_k(2I - AY_k). \quad k = 0, 1, \ldots.$$

Ben-Israel and Cohen realized that Newton's iterative method works not only for usual matrix inversion but also for the Moore-Penrose inversion. In order to prove this finding, first they proved the convergence of (4.12) to the pseudoinverse and after this they managed to prove (4.20). Therefore Newton's iteration process with the starting value (4.13) is convergent to the Moore-Penrose inverse.

Consider 5.2. By Theorem (4.2.7) The optimal α is, $\alpha \approx 0.013128302506547$. For $\varepsilon = 0.0000005$ we have the following chart



On the chart horizontal *axis* describes the number of iteration steps and vertical *axis* shows the matrix norm 1 of $X_n - A^+$. For $\alpha \approx 0.013128302506547$ after 14 steps we get the following approximation of the pseudoinverse of A with the precision $\|X_{14} - X_{13}\| = 1.292523588158900 \cdot 10^{-9}$. We got:

$$\begin{pmatrix} 0.888888888888889 & -0.870370370370370 & -0.259259259259259 & 1.129629629629630 \\ -0.444444444444444 & 0.518518518518519 & -0.037037037037037 & -0.481481481481481 \\ 0.444444444444444 & -0.351851851851851 & -0.296296296296296 & 0.648148148148148 \\ -0.333333333333333 & 0.388888888888890 & 0.222222222222223 & -0.611111111111110 \end{pmatrix}$$

Making test for different α showed that the number of the iteration steps needed is within the same order of magnitude(in our case < 30).

α	ε	Norm	Iteration number
0.013129	0.0000005	1	15
0.003129	0.0000005	1	17
0.000129	0.0000005	1	22
0.000029	0.0000005	1	24
0.000009	0.0000005	1	26

5.2 Investigation of the Iterative Method (4.27)

The following iterative method is based on (4.14).

$$x_{n+1} = x_n - \lambda_n a^*(ax_n - c).$$

Truly, from (4.14) we obtain:

$$Y_{k+1} = Y_k - Y_k(AY_k - I). \tag{5.3}$$

We can notice that there exist $C_k \in \mathbb{C}^{n\times n}$ for $\forall k$ such that the following holds

$$Y_{k+1} = Y_k - C_k A^*(AY_k - I),$$

where α satisfies (4.11). First by induction we will show $Y_k = C_k A^*$. Truly, for $k = 1$:

$$Y_1 = Y_0(2I - AY_0) = \alpha A^*(2I - \alpha AA^*) = \alpha(2I - \alpha A^*A)A^* = C_1 A^*,$$

where $C_1 \equiv \alpha(2I - \alpha A^*A)$. If we assume for k that $Y_k = C_k A^*$, then we get:

$$Y_{k+1} = Y_k(2I - AY_k) = C_k A^*(2I - AC_k A^*) = (2C_k - C_k A^* AC_k)A^* = C_{k+1}A^*,$$

where $C_{k+1} = 2C_k - C_k A^* AC_k$. Therefore from (5.3) it follows:

$$Y_{k+1} = Y_k - C_k A^*(AY_k - I). \tag{5.4}$$

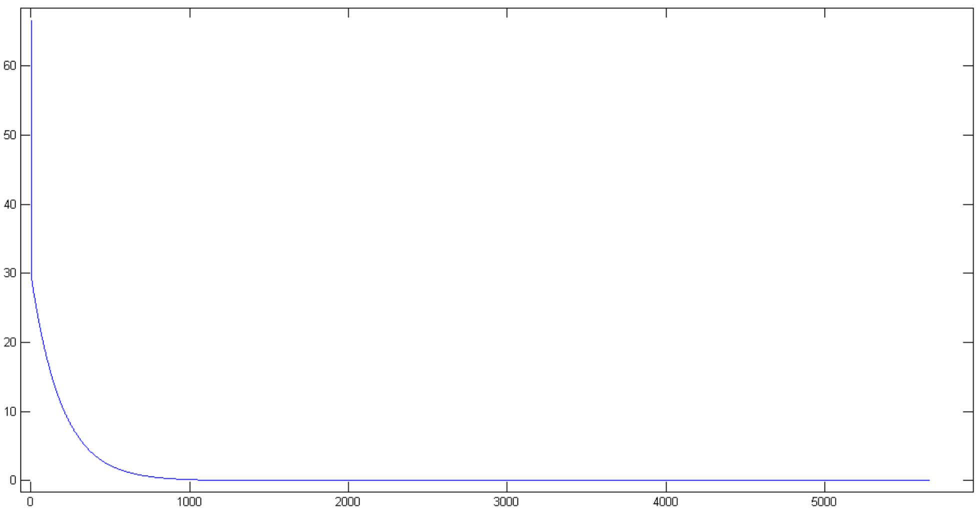
In 4.27 instead of C_k the multiplier is $\lambda_k I$. This substitution simplifies calculation of the iterative process, because instead of calculating some complicated matrices we just take scalars which economizes the computational time and space of a computer's memory.

Consider (5.2). For constant $\lambda_n = 0.013111605$ after 3873 steps we get the following approximation for the Pseudoinverse of A with the precision $\|X_{3874} - X_{3873}\| = 4.978585573800487 \cdot 10^{-7}$.

$$\begin{pmatrix} 0.888873326648520 & -0.870312663934795 & -0.259281212930094 & 1.129547299243109 \\ -0.444437050781649 & 0.518491092310388 & -0.037026627204830 & -0.481442372790331 \\ 0.444436275866865 & -0.351821571624350 & -0.296307840134912 & 0.648104926452780 \\ -0.333325834733609 & 0.388861048819393 & 0.222232728132671 & -0.611071464311854 \end{pmatrix}$$

By observing results we see that (4.27) needs more iteration steps to compute pseudoinverse than (4.14) but this does not mean that the former is slower than the latter. As we see in (5.3) for each iteration step it needs to compute two matrix multiplications in contrast to (5.4) when it needs one multiplication for computing one iteration. Therefore if we have a quite big dimensional matrix then for (5.4) computation a computer will need less time and memory space then for (5.3).

For $\lambda_n = 0.013 - 1/n^7$ we get:



As observation showed if we accelerate λ_n to 0 (for example $\lambda_n = \frac{1}{n^{15}}$) then the iteration process is not convergent to A^+, it is convergent to the starting value X_0. Which contradicts to the theorem(4.3.4).

5.3 Investigation of the Iterative Method (4.28)

We see that the difference between (4.28) and (5.4) is that instead of C_k we take $(\lambda_n I + A^*A)^{-1}$.

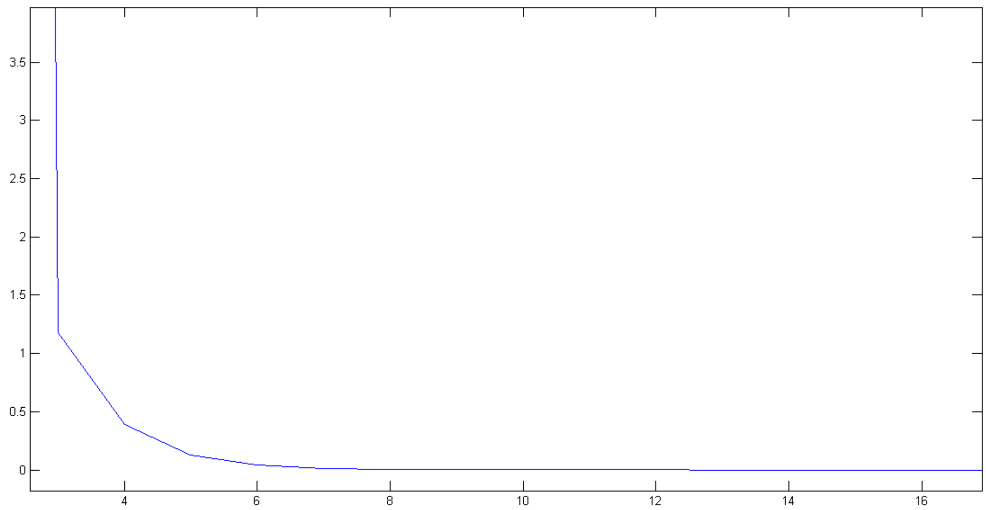
$$x_{n+1} = x_n - (\lambda_n I + a^*a)^{-1}a^*(ax_n - c).$$

This makes sense, because as we know

$$\lim_{k\to\infty} C_k A^* = A^+. \tag{5.5}$$

Let $Y_n = (\lambda_n I + A^*A)^{-1}A^*$, as known $\lim_{n\to\infty} Y_n = A^+$ if $\lambda_n \to 0$.
As results showed, for all A we can find very small $(\lambda_n)_n$ such that $(\lambda_n I + A^*A)^{-1}$ is not ill-conditioned for all n and Y_n converges to A^+ much quicker then $C_k A^*$. Therefore substituting Y_n into (5.4) instead of $C_k A^*$ causes acceleration of the iteration process.

Consider 5.2. For constant sequence $\lambda_n = 0.1$ we get:



For $\lambda_n = 0.1$ after 14 steps we get the following approximation for the pseudoinverse of A with precision $\|X_{14} - X_{13}\| = 2.475861949347014 \cdot 10^{-7}$.

$$\begin{pmatrix} 0.888888879447414 & -0.870370335354776 & -0.259259272566503 & 1.129629579684490 \\ -0.444444439961510 & 0.518518501892622 & -0.037037030718536 & -0.481481457766725 \\ 0.444444439485965 & -0.351851833462300 & -0.296296303285054 & 0.648148121917767 \\ -0.333333328793601 & 0.388888872052371 & 0.222222228620744 & -0.611111087096000 \end{pmatrix}$$

In the following table we consider the case when the $(\lambda_n)_n$ sequence is constant.

λ_n	ε	Norm	Iteration number
100	0.0000005	1	4989
100	0.000005	1	3835
10	0.0000005	1	620
10	0.000005	1	504
1	0.0000005	1	80
1	0.000005	1	68
0.1	0.0000005	1	16
0.1	0.000005	1	14
0.0001	0.0000005	1	4
0.00001	0.0000005	1	3

If we change the matrix order investigation significantly, for example to consider A/1000, then the number of iterations may change largely, but the property of the iterative method remains. Taking $(\lambda_n)_n$ closely to 0 reduces the number of iterations. But as we mentioned above for parameter $(\lambda_n)_n$ we should make some distance to 0 in order to avoid ill-conditioning.

5.4 Investigation of the Iterative Method (4.4)

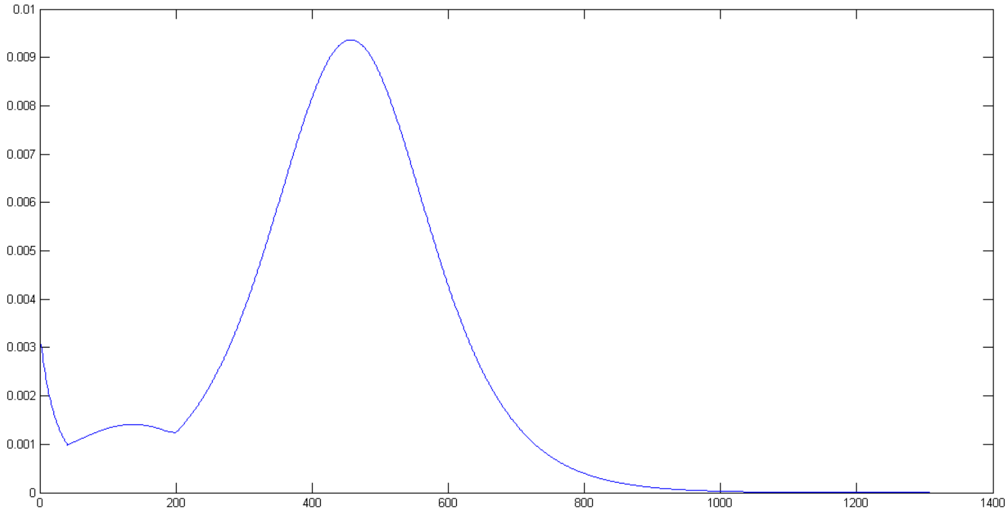
For the iterative method

$$X_{k+1} = (I - \beta X_k A)X_k + \beta X_k = (1+\beta)X_k - \beta X_k A X_k,$$

consider (5.2). For $\beta = 0.013128318235738$ after 1307 steps we get the following approximation for the Pseudoinverse of A with the precision $\|X_{1307} - X_{1306}\| = 4.987680035095643 \cdot 10^{-7}$.

$$\begin{pmatrix} 0.888877558528980 & -0.870359076900360 & -0.259255442025780 & 1.129614519686639 \\ -0.444439056002752 & 0.518513137432797 & -0.037038817317009 & -0.481474319355279 \\ 0.444438502526237 & -0.351845940228077 & -0.296294260103299 & 0.648140199570867 \\ -0.333327889817448 & 0.388883472112553 & 0.222220365640064 & -0.611103837752616 \end{pmatrix}$$

For $\beta = 0.013128318235738$ we get:



α	ε	Norm	Iteration number
0.000229	0.0000005	1	75188
0.003129	0.0000005	1	5489
0.003129	0.000005	1	4754
0.013129	0.0000005	1	1307
0.013129	0.000005	1	1133
0.06	0.0000005	1	284
0.06	0.000005	1	247
0.08	0.0000005	1	212
0.08	0.000005	1	185

5.5 Investigation of the Iterative Method (4.30)

As we saw in the example at the introduction part of my thesis, a computer can make some computational errors in some special situations. In this example a computer error significantly impacts on the results. Therefore, as we already mentioned, in order to avoid it, iterative methods which compute a problem in a different way were invented. As we saw above, all iteration methods presented in this thesis converge to a real solution of a problem, but we cannot be sure that they work well when we apply computer programming languages in practice. The authors of [2] describe the other method

$$x_{n+1} = x_n - (\lambda_n + a^* a)^{-1} a^* (a x_n - c).$$

Although they do not state it, it is in fact a method that takes into consideration the computational errors when implementing their previous method (4.28). If we take $c = Identity$, we get

$$\begin{aligned} x_{n+1} &= x_n - (\lambda_n + a^*a)^{-1}a^*(ax_n - I) \\ &= x_n - (\lambda_n + a^*a)^{-1}((a^*a + \alpha_n)(a^*a + \alpha_n)^{-1}a^*ax_n - a^*) \\ &= x_n - (\lambda_n + a^*a)^{-1}((a^*a + \alpha_n)y_n ax_n - a^*), \end{aligned}$$

where $y_n = (a^*a+\alpha_n)^{-1}a^*$. It is well known that $\lim_{n\to\infty} y_n = a^+$ when $\alpha_n \longrightarrow 0$. Therefore $y_n - a^+ = b_n$ is small. If we consider case $a^+ax_0 = x_0$ (consequently $a^+ax_n = x_n$), we obtain:

$$y_n ax_n = (a^+ + b_n)ax_n = a^+ax_n + b_n ax_n \approx a^+ax_n + 0 = x_n.$$

If we substitute this into the above equation, we get:

$$\begin{aligned} x_{n+1} &= x_n - (\lambda_n + a^*a)^{-1}((a^*a + \alpha_n)x_n - a^*) \\ &= x_n - (\lambda_n + a^*a)^{-1}(a^*ax_n - a^* + \alpha_n x_n). \end{aligned}$$

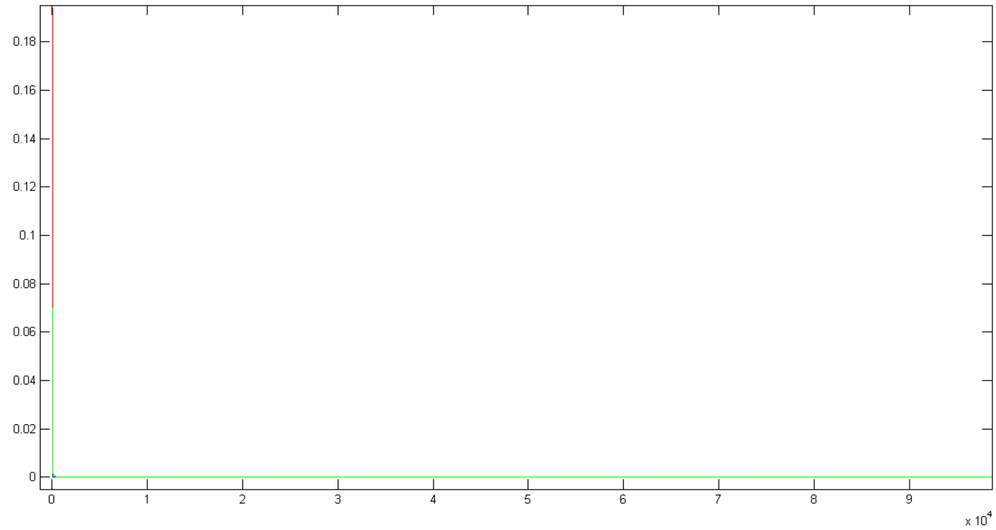
In this we recognise the method (4.30), where $(\beta_n)_n$ is a bounded sequence, $(\alpha_n)_n$ is a decreasing sequence to 0 and $\beta_n - \alpha_n > 0, \quad \forall n$

In the proof part of the Theorem 4.3.7. we saw that

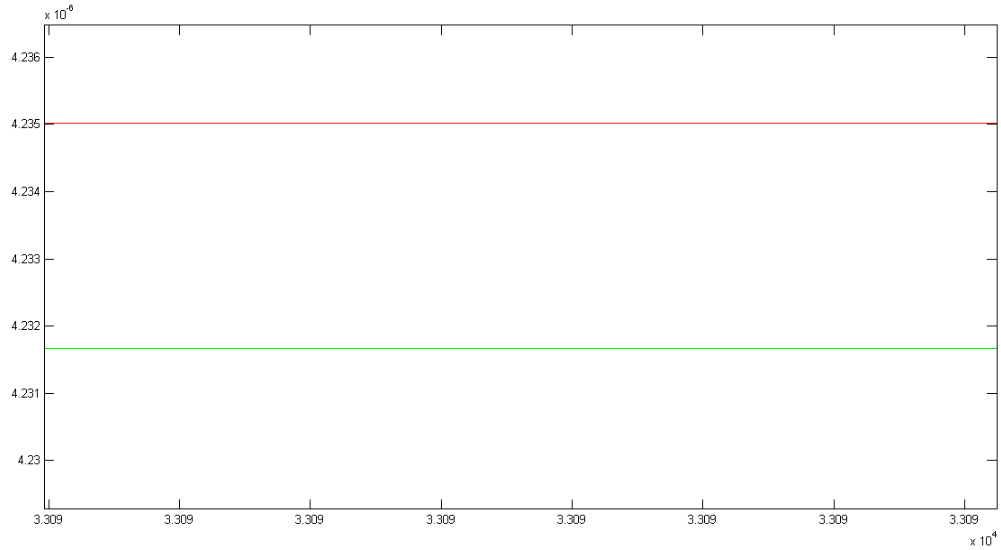
$$\|x_{n+1} - a^+\| \leq \|x_{n+1} - y_n\| + \|y_n - a^+\|.$$

Let us observe charts of each part of the previous inequality.

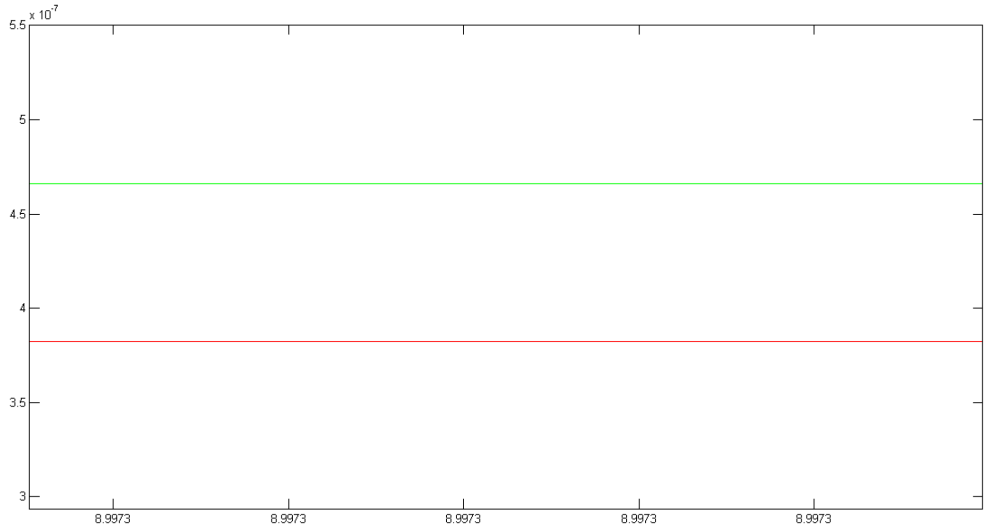
Consider the matrix (5.2). For $\beta_n = 0.01$ and $\alpha_n = 0.01/n, \quad n > 1$ we obtain



Red line describes $\|x_{n+1} - a^+\|$ and blue line $\|y_n - a^+\|$. Let us magnify any area of the chart. We see



If we accelerate α to zero, $\alpha_n = 1/n^6$ and we magnify the chart again, we observe that on both charts the difference between the graph of $\|x_{n+1} - a^+\|$ and the graph of $\|y_n - a^+\|$ is very small.

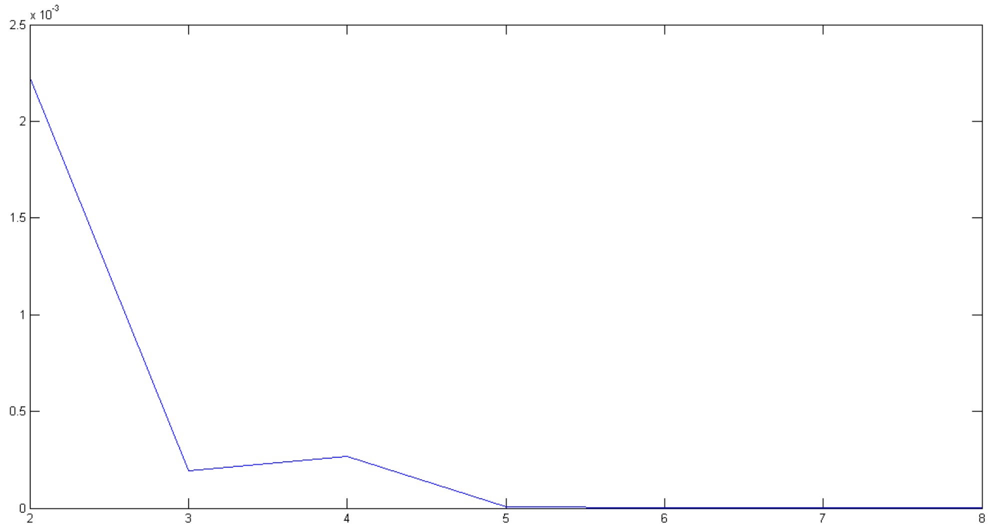


Therefore, by logic, instead of observing the complicated iterative method (4.30) with 2 parameters, we simply can take the following process:

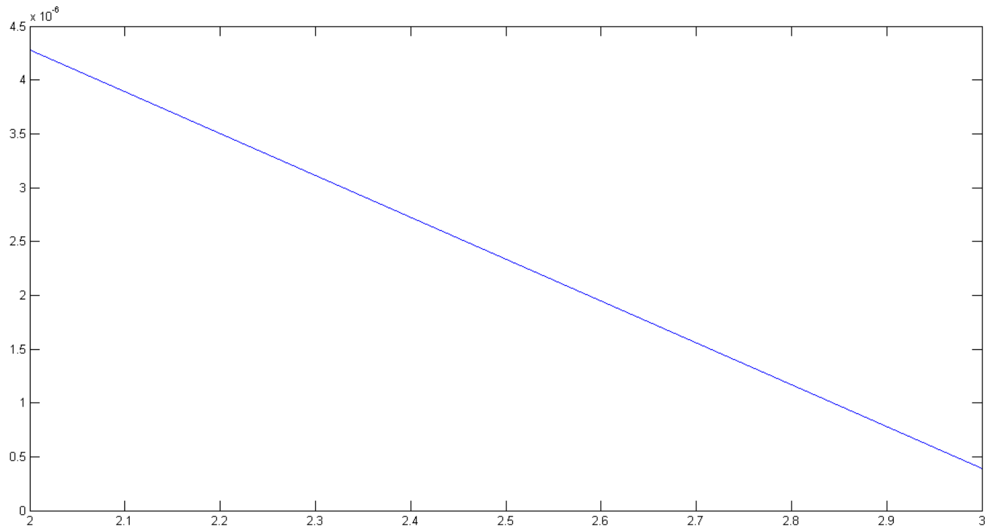
$$y_n = (a^*a + \alpha_n)^{-1}a^*.$$

But a problem might occur during the computation of y_n. If the iterative method's approximative value is very small then $(a^*a + \alpha_n)$ might become ill-conditioned for quite small α_n. Therefore in order to avoid it, we can use an iterative method for computing matrix inversion. The above described Gauss-Seidel's iterative method is less productive because when $\alpha_n \to 0$ and hence $(a^*a + \alpha_n)$ becomes ill-conditioned, then it increases the number of iterations needed.

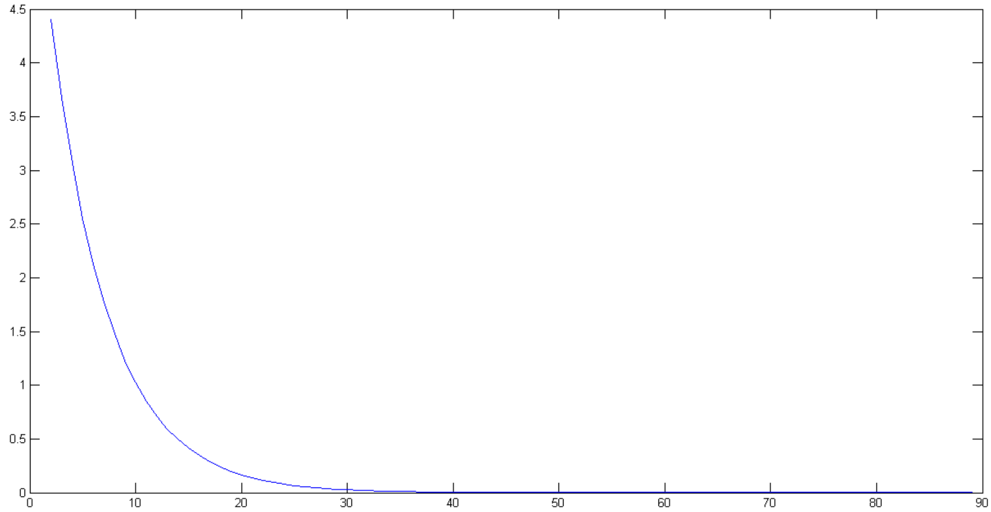
However if we apply Newton's method for computing $(a^*a+\alpha_n)^{-1}$, we obtain better results.
For $\alpha_n = \frac{1}{n^6}$, $\varepsilon = 0.0000005$ we get



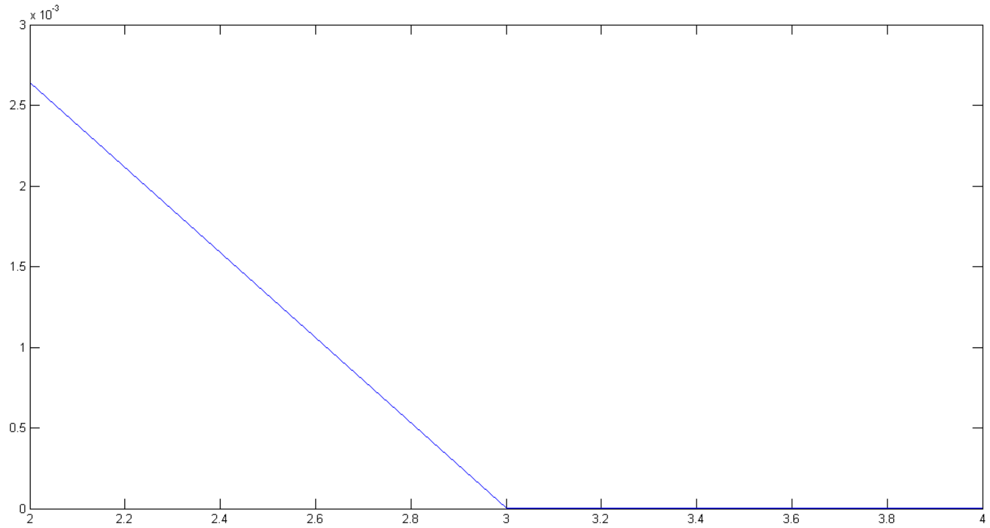
For $\alpha_n = \frac{1}{n^{15}}, \quad \varepsilon = 0.0000005$



For $\beta_n = 1, \quad \alpha_n = \frac{1}{n^{15}}$



For $\beta_n = 0.0001, \quad \alpha_n = \frac{1}{n^{15}}$



We see that when β_n becomes closer to zero then the iteration process needs fewer steps for computing A^+. But generally we cannot reduce it too much, because for computing (4.30) we need to invert $(\beta_n + A^*A)$ and it might become an ill-conditioned matrix. Therefore generally β_n should be bounded away from 0.

Chapter 6

Conclusion

In this thesis we presented a definition of the Moore-Penrose inverse, its usage, as well as two classic methods of its computation. In addition, we described a situation when computing the pseudoinverse through these methods by using computer programs is not correct. After this we presented different iteration processes for computing the Moore-Penrose inverse, we compared them to one another, explained their differences and made analysis to each of them. The numerical results and charts presented in the thesis were derived by using program Matlab.

It can be recommended that future research should focus on more investigation of each iterative method. As we mentioned in the previous chapter, the iterative process (4.27) does not work in the case when the sequence of parameters goes to 0 very quickly. Which contradicts to the theorem(4.3.4). Therefore, future work can improve this method, also to find optimal parameter for (4.4).

Chapter 7

Bibliography

[1] Adi Ben-Israel and Dan Cohen. On iterative computation of generalized inverses and associated projections. *SIAM Journal on Numerical Analysis*, 3(3):pp. 410–419, 1966.

[2] Dragan S. Djordjevic and Predrag S. Stanimirovic. Iterative methods for computing generalized inverses related with optimization methods. *The Journal of the Australian Mathematical Society*, (78):257 – 272, 2005.

[3] Wolf Holzmann. Ill-conditioned matrices. http://www.cs.uleth.ca/ holzmann/notes/illconditioned.pdf.

[4] Victor Pan and Robert Schreiber. An improved newton iteration for the generalized inverse of a matrix, with applications. *SIAM J. Sci. Stat. Comput*, 12:1109–1131, 1991.

[5] Marko D. Petkovic and Predrag S. Stanimirovic. Iterative method for computing the moore-penrose inverse based on penrose equations. *Journal of Computational and Applied Mathematics*, 235(6):1604 – 1613, 2011.

[6] R. Piziak and P.L. Odell. *Matrix Theory: From Generalized Inverses to Jordan Form.* Pure and Applied Mathematics. Chapman & Hall/CRC, 2007.

[7] S. Simons. 80.47 an iterative algorithm for matrix inversion which is always convergent. *The Mathematical Gazette*, 80(489):pp. 567–569, 1996.

[8] Torsten Soderstrom and G. W. Stewart. On the numerical properties of an iterative method for computing the moore- penrose generalized inverse. *SIAM Journal on Numerical Analysis*, 11(1):pp. 61–74, 1974.