Scalar correction method for finding least-squares solutions on Hilbert spaces and its applications

Sladjana Miljković[1], Marko Miladinović[2]*, Predrag S. Stanimirović[3], Dragan S. Đorđević[4]

[1,2,3,4] *University of Niš, Faculty of Sciences and Mathematics,*
Višegradska 33, 18000 Niš, Serbia. †

E-mail: [1]slagana256@yahoo.com, [2]markomiladinovic@gmail.com,
[3]pecko@pmf.ni.ac.rs, [4]dragan@pmf.ni.ac.rs

Abstract

We use the idea of two-point stepsize gradient methods, developed to solve unconstrained minimization problems on $\mathbb{R}^n$, for computing least-squares solutions of a given linear operator equation on Hilbert spaces. Among them we especially pay attention to corresponding modification of the scalar correction method. An application of this approach is presented related to computation of $\{1,3\}$ inverses and the Moore-Penrose inverse of a given complex matrix. Convergence properties of the general gradient iterative scheme for computation of various pseudoinverses are investigated. The efficiency of the presented algorithm is theoretically verified and approved by selected test matrices.

AMS Subj. Class.: 90C30,15A09

Keywords: Generalized inverses, unconstrained optimization, least-squares solutions, gradient method, convergence rate.

1 Introduction

Solving the system of linear operator equations or finding a generalized inverse of a given operator $A \in \mathcal{L}(\mathcal{H},\mathcal{K})$, where $\mathcal{L}(\mathcal{H},\mathcal{K})$ denotes the space of linear bounded operators between Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$, is an interesting problem. Many different techniques are developed to solve these problems. One of the most interesting investigation fields is the usage of optimization methods in order to find some generalized inverses. This approach includes methods based on the first or on the second order optimization, at first introduced for functions on $\mathbb{R}^n$ and later extended to some more general spaces. Many algorithms of this type can be found in [4, 12, 15, 21, 22, 28].

The equation $Ax = b$, $b \in \mathcal{K}$ may or may not have a solution, depending on whether b is in the range $\mathcal{R}(A)$ of A or not. Even if $b \in \mathcal{R}(A)$ the solution need not to be unique. In cases where $b \notin \mathcal{R}(A)$ or the solution is not unique, it is possible to compute least-squares solutions which minimize the quadratic functional $q(x) = \frac{1}{2}\|Ax - b\|^2$.

Nashed in [22] minimized the functional $q(x)$ in order to find a solution of the operator equation $Ax = b$, where $A \in \mathcal{L}(\mathcal{H},\mathcal{H})$ is such that $\mathcal{R}(A)$ is closed. The minimization of the functional $q(x)$ is accomplished by using the iterative scheme

$$x_{k+1} = x_k - \alpha_k r_k, \tag{1.1}$$

[*] Corresponding author

[†] The authors gratefully acknowledge support from the research project 144011 of the Serbian Ministry of Science.

where

$$r_k = A^*Ax_k - A^*b \quad \text{and} \quad \alpha_k = \frac{\|r_k\|^2}{\|Ar_k\|^2}.$$

Since this method is actually the steepest descent method, the stepsize α_k is chosen according to the strategy that guarantees the most rapid decrease of $\|Ax_{k+1} - b\|$. The linear convergence of the method to least-squares solutions of the equation $Ax = b$ is established in [22], for an arbitrary initial approximation $x_0 \in \mathcal{H}$. Additionally, it is shown that the steepest descent method converges to the minimum-norm least-squares solution, i.e. $\lim_{k\to\infty} x_k = A^\dagger b$, if and only if $x_0 \in \mathcal{R}(A^*)$.

Inspired by the Nashed steepest descent method, the authors in [12] considered the iterative scheme of the same form in more general spaces, $\mathcal{C}^*$ algebras. The established proof for the linear convergence of the method differs from the one from [22], not only with respect to the observed spaces but also with respect to the used norms. Namely, the presented convergence theorem in [12] is given in terms of an operator norm while the convergence of the Nashed steepest descent scheme is given in terms of the Frobenius norm.

Despite the optimal property, the steepest descent method behaves poorly, except for very well conditioned functions and converges slowly (see [1, 14]). In some particular cases, such as quadratic functions, it is possible to compute the steplength γ_k analytically. Even for these functions the steepest descent method behaves increasingly badly when the condition number of the matrix deteriorates. The authors in [26] stressed out that the poor behavior of the steepest descent method is due to the optimal Cauchy choice of the steplength γ_k and not to the choice of the search direction (direction of the negative gradient). For this purpose we, especially, pay attention to the steplength calculation algorithm which searches along the direction of the negative gradient.

The two-point stepsize gradient method introduced in [2] is proved to be more effective and thus, preferable over the classical steepest descent method both in theoretical investigations and in real computations. This method proposed by Barzilai and Borwein (called BB method) is also gradient method which uses a different strategy for choosing the steplength α_k. The stepsize along the negative gradient direction, in BB method, is computed from a two-point approximation to the secant equation required in quasi-Newton methods. There are a number of papers dealing with the convergence properties of the two-point stepsize methods as well as some advantages of these methods with respect to some other optimization methods (see [8, 9, 10, 13, 25]).

Additionally, some other optimization methods, such as the conjugate gradient methods as well as variable metric methods have been used in the minimization of the functional $q(x)$ and in solving the matrix equations $AX = I_m,\ XA = I_n$, where I_m and I_n are appropriate identity matrices (see, for example [15, 16, 21, 28]). Also, interesting methods for computing the Moore-Penrose inverses and Drazin inverses of Toeplitz matrices are proposed in [6, 29]. These methods belongs to the group of Newton's and modified Newton's iteration. Toeplitz matrices, by itself, are very interesting and have been studied much, recently. These special type of structured matrices will be covered later in section Numerical results.

Main idea used in [20, 27] is approximation of the Hessian by an appropriate scalar matrix, according to the classification of the quasi-Newton methods from [5]. Following this approach, the iterative scheme

$$x_{k+1} = x_k - \gamma_k \nabla q(x_k), \quad k \geq 0, \tag{1.2}$$

can be considered as a variant of the quasi-Newton method, where the symmetric positive definite matrix B_k is defined as a scalar matrix. In the paper [20] the authors introduced an algorithm of this type, called the the scalar correction method (SC method). Very useful characteristics of SC algorithm, aimed to solve unconstrained nonlinear optimization problems, such as the simplicity, the efficiency and very low memory space requirements are verified in [20] through a set of large scale test examples. Therefore, it seems realistic to expect that the characteristics of the SC method, observed in [20], will remain in minimization of the functional $q(x)$.

In order to obtain an algorithm with better performances, the main goal in the present paper will be the usage of two-point step size gradient methods introduced in [2, 20] for obtaining a least-squares solution of a given operator equation $Ax = b$. Special attention will be dedicated to SC method from [20]. Later, similarly as in [12], the Moore-Penrose inverse of a given matrix will be determined, observing the iterative process

$$X_{k+1} = X_k - \gamma_k A^*(AX_k - I), \quad k \geq 0, \tag{1.3}$$

The paper is organized as follows. In the next section we state some known definitions and results related to a Frèchet derivative of an operator on a Hilbert space. The extension of SC method from [20] with regard to Hilbert spaces is presented in the third section. Besides, we introduse an algorithm for finding least-squares solutions of an operator equation $Ax = b$, where $A \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ is chosen such that $\mathcal{R}(A)$ is closed. Also, in this section we establish the linear convergence of the algorithm. An iterative process for computing $\{1,3\}$-inverses, and particularly the Moore-Penrose inverse, of an arbitrary complex matrix, is derived as an application of the given algorithm in the section four. We also consider the correspondent modification of BB method. In Section 5 convergence properties of general gradient methods for computing least-squares solutions and the minimum-norm least-squares solution of the operator equation $Ax = b$ are presented. In addition, the convergence result of the iterative process for computing $\{1,3\}$-inverses (including the Moore-Penrose inverse) is considered. Appropriate numerical results can be found in the last section.

2 Preliminaries

For the sake of completeness, we restate main known facts about the Fréchet derivative from [7, 17].

Definition 2.1. *Let $\mathcal{H}$ and $\mathcal{K}$ be Hilbert spaces and $\mathcal{U} \subset \mathcal{H}$ is an open set. Let $f : \mathcal{U} \to \mathcal{K}$ be an operator and $x \in \mathcal{U}$. If there is a bounded linear operator $g : \mathcal{H} \to \mathcal{K}$ such that*

$$\lim_{\|h\| \to 0} \frac{\|f(x+h) - f(x) - g(h)\|}{\|h\|} = 0,$$

$h \in \mathcal{H}$, we say that f is Fréchet differentiable at x, or simply differentiable at x; g is called the (Fréchet) derivative of f at x and we will denote it by $Df(x) \in \mathcal{L}(\mathcal{H}, \mathcal{K})$.

Definition 2.2. *An operator f is $(n+1)$-times differentiable on $\mathcal{U}$ if it is n times differentiable on $\mathcal{U}$ and for each x in $\mathcal{U}$ there exists a continuous multilinear map g of $(n+1)$ arguments such that the limit*

$$\lim_{\|h_{n+1}\| \to 0} \frac{\|D^n f(x + h_{n+1})(h_1, \dots, h_n) - D^n f(x)(h_1, \dots, h_n) - g(h_1, \dots, h_{n+1})\|}{\|h_{n+1}\|} = 0 \tag{2.1}$$

exists uniformly for $h_1, h_2, \dots, h_n$ in bounded sets in $\mathcal{H}$. In this case, g is the $(n+1)$st derivative of f at x.

Proposition 2.1. *Let D be a convex subset of $\mathcal{H}$ and f is $(n+1)$-times Frechet differentiable operator on D. Then if x and $x+p$ are given elements in D we have*

$$f(x+p) = \sum_{k=0}^{n} \frac{1}{k!} D^{(k)} f(x) \underbrace{(p, p, \dots, p)}_{k \ times} + w(x,p)$$

where

$$\|w(x,p)\| \leq \frac{1}{(n+1)} \sup_{t \in [0,1]} \|D^{(n+1)} f(x + tp)\| \|p\|^{n+1}.$$

Definition 2.3. [18] *Let $\mathcal{H}$ be Hilbert space, $\mathcal{U} \subset \mathcal{H}$ is an open set and $f : \mathcal{U} \to \mathbb{R}$ is a given differentiable functional. The gradient of the functional f is the linear map*

$$\nabla f : \mathcal{U} \to \mathcal{H} \quad \text{such that} \quad \langle \nabla f(x), h \rangle = Df(x)(h),$$

where $Df(x)(h)$ means the linear map $Df(x)$ applied to the vector $h \in \mathcal{H}$.

The existence and uniqueness of such linear map follows straight from the application of the Riesz representation theorem of the linear bounded operator $Df(x) : \mathcal{H} \to \mathbb{R}$.

Definition 2.4. *Let $\mathcal{H}$ be Hilbert space, $\mathcal{U} \subset \mathcal{H}$ is an open set and $f : \mathcal{U} \to \mathbb{R}$ is a given twice differentiable functional. The Hessian of the functional f is the linear operator*

$$\nabla^2 f \in \mathcal{L}(\mathcal{U} \times \mathcal{H}, \mathcal{H}) \quad \textit{such that} \quad \langle \nabla^2 f(x,p), h \rangle = D^2 f(x)(p,h),$$

where $D^2 f(x)(p,h)$ means the linear map $D^2 f(x)(p)$ applied to the vector $h \in \mathcal{H}$.

The existence and uniqueness of such linear map follows straight from the Riesz representation theorem for the linear bounded operator $D^2 f(x)(p) : \mathcal{H} \to \mathbb{R}$.

3 Scalar correction method for solving linear operator equations

Let $\mathcal{H}$ and $\mathcal{K}$ be given Hilbert spaces and $A \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ be given operator such that $\mathcal{R}(A)$ is closed. We minimize the functional $q : \mathcal{U} \to \mathbb{R}$ defined by

$$q(x) = \frac{1}{2}\|Ax - b\|^2 = \frac{1}{2}\langle Ax - b, Ax - b\rangle. \tag{3.1}$$

on an open set $\mathcal{U} \subset \mathcal{H}$. For that purpose we analyze the gradient iterative scheme (1.2) where $\gamma_k > 0$ is a stepsize. Taking into account that $g_k = \nabla q(x_k) = A^*(Ax_k - b)$, which is not difficult to show from (3.1), we consider an iterative process given in the following general form

$$x_{k+1} = x_k - \gamma_k A^*(Ax_k - b) \tag{3.2}$$

for the purpose of finding least-squares solutions of the equation $Ax = b$. The importance in choosing appropriate stepsize in order to obtain convergence as well as good computational performance is obvious.

In the rest of this section we use the stepsize determined according to the idea of the scalar correction method, introduced in [20]. For the sake of completeness we extend the basic ideas for SC method with respect to Hilbert spaces. In order to obtain an appropriate stepsize we use the information of the stepsize obtained in the previous step and try to correct it by adding some scalar. To determine that scalar properly, the idea of a two-point approximation to the secant equation is used, similarly as in BB method. The final choice for the stepsize is done by relaxing the stepsize as much as it is possible in view of two additional steplengths which are also obtained from the first order secant equation.

That is, first we have to obtain the analog to the secant equation for an operator on a Hilbert space. Based on Proposition 2.1 we have that the first order Taylor series expansion for the operator Dq is

$$Dq(x) = Dq(x_{k+1}) + D^2 q(x_{k+1})(x - x_{k+1}) + w(x, x - x_{k+1}), \tag{3.3}$$

which is equivalent to

$$Dq(x)(h) = Dq(x_{k+1})(h) + D^2 q(x_{k+1})(x - x_{k+1}, h) + w(x, x - x_{k+1})(h) \quad \text{for all} \;\; h \in \mathcal{H}.$$

From Definition 2.3 and Definition 2.4 we have

$$\langle \nabla q(x) - \nabla q(x_{k+1}), h \rangle - \langle \nabla^2 q(x_{k+1}, x - x_{k+1}), h \rangle = w(x, x - x_{k+1})(h) \quad \text{for all} \;\; h \in \mathcal{H}.$$

Taking $h = \nabla q(x) - \nabla q(x_{k+1}) - \nabla^2 q(x_{k+1}, x - x_{k+1})$ and having in mind that $w(x, x - x_{k+1})$ stands for second order residual we get

$$\nabla q(x) - \nabla q(x_{k+1}) \approx \nabla^2 q(x_{k+1})(x - x_{k+1}).$$

Setting $x = x_k$, $g_k = \nabla q(x_k)$, $H_k = \nabla^2 q(x_k)$, we get

$$H_{k+1}(s_k) \approx y_k, \tag{3.4}$$

where

$$s_k = x_{k+1} - x_k, \quad \text{and} \quad y_k = g_{k+1} - g_k.$$

Now, we want to approximate H_{k+1} with some identical operator $I : \mathcal{H} \to \mathcal{H}$ multiplied by the real parameter $\frac{1}{\gamma_{k+1}}$ such that $H_{k+1}(h) = \frac{1}{\gamma_{k+1}} h$ holds for all $h \in \mathcal{H}$.

The main problem is to compute γ_k, $k \geq 0$. Let us suppose that we have $\gamma_0 = 1$. Since we have the initial value γ_0 we observe the following correction

$$\gamma_{k+1} = \gamma_k + a_k, \quad k > 0. \tag{3.5}$$

in order to find the stepsize γ_{k+1} for the next iteration. The method of computing the scalar correction a_k in each iteration and therefore a new step-length γ_{k+1} is presented in detail in the paper [20]. We give here only the final expression

$$\gamma_{k+1}^{SC} = \begin{cases} \frac{\langle s_k, r_k\rangle}{\langle y_k, r_k\rangle}, & \langle y_k, r_k\rangle > 0 \\ \frac{\|s_k\|}{\|y_k\|}, & \langle y_k, r_k\rangle \leq 0 \end{cases}, \quad k \geq 0, \tag{3.6}$$

where $r_k = s_k - \gamma_k y_k$.

Corresponding algorithm is defined as follows.

Algorithm 3.1 SC method for computing least-squares solutions

```
Require: An operator $A : \mathcal{H} \to \mathcal{K}$ such that $\mathcal{R}(A)$ is closed, chosen initial point $x_0 \in \mathcal{H}$ and real
    positive constants $0 < \varepsilon \ll 1$, $0 < \xi_1 \ll \frac{2(1-\varepsilon)}{\|A\|^2}$.
1: Set $k = 0$, compute $q(x_0)$, $g_0$ and use $\gamma_0 = 1$.
2: If test criteria are fulfilled then go to Step 7; otherwise, go to the next step.
3: Compute $x_{k+1}$ using (3.2), $q(x_{k+1})$, $g_{k+1}$, $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.
4: Determine $\xi_2^{(k+1)} = 2(1-\varepsilon)\frac{\|g_{k+1}\|^2}{\|Ag_{k+1}\|^2}$.
5: Compute the stepsize $\gamma_{k+1}$ using (3.6). If $\gamma_{k+1} < \xi_1$ or $\gamma_{k+1} > \xi_2^{(k+1)}$, set $\gamma_{k+1} = \xi_2^{(k+1)}$.
6: Set $k := k + 1$ and go to Step 2.
7: Return $x_{k+1}$ and $q(x_{k+1})$.
```

Proposition 3.1. *Algorithm* 3.1 *is well defined, i.e. the interval* $(\xi_1, \xi_2^{(k)})$, $k \geq 1$ *is non-empty, for a chosen real constant* ξ_1. *Additionally, the sequence of stepsizes* $(\gamma_k)_k$ *is positive and it is bounded by real constants.*

Proof. Since

$$\xi_2^{(k)} = 2(1-\varepsilon)\frac{\|g_k\|^2}{\|Ag_k\|^2} \geq \frac{2(1-\varepsilon)}{\|A\|^2}$$

one can always choose constant ξ_1 such that

$$0 < \xi_1 \ll \frac{2(1-\varepsilon)}{\|A\|^2} \leq \xi_2^{(k)}.$$

Thus, the interval $(\xi_1, \xi_2^{(k)})$ is non empty in each iteration.

Taking into account that $\|Ax\| \geq j(A) \cdot \|x\|$, where $j(A) = \inf_{\|x\|=1} \|Ax\|$, it is not difficult to show that $\xi_2^{(k)} \leq 2 \cdot j(A)^{-2}$, $k \geq 1$. Thus, the following sequence of inequalities holds

$$0 < \xi_1 \leq \gamma_k \leq \xi_2^{(k)} \leq 2 \cdot j(A)^{-2}, \quad k \geq 1,$$

which is a verification of the second statement. □

The following theorem shows the convergence of the method given by Algorithm 3.1 and also characterizes the set of least-squares solutions of the operator equation $Ax = b$. An auxiliary result is used.

Proposition 3.2. [22, 23] If $\mathcal{R}(A)$ is closed then the set S of all least-squares solutions of the system $Ax = b$ is given by

$$S = A^{\dagger}b \oplus \mathcal{N}(A) = \{A^{\dagger}b + (I - A^{\dagger}A)y |\ y \in \mathcal{H}\},$$

where $\mathcal{N}(A)$ denotes the null space of A.

Theorem 3.1. *Let $\mathcal{H}$ and $\mathcal{K}$ be given Hilbert spaces and $A \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ be an operator such that $\mathcal{R}(A)$ is closed. The sequence $(x_k)_k$ determined by Algorithm* 3.1 *converges to a least-squares solution of the equation $Ax = b$.*

For an arbitrary initial approximation $x_0 \in \mathcal{H}$ the limit $\lim_{k\to\infty} x_k$ satisfies

$$\lim_{k\to\infty} x_k = A^{\dagger}b + (I - A^{\dagger}A)x_0.$$

Additionally, $x_0 \in \mathcal{R}(A^*)$ if and only if $\lim_{k\to\infty} x_k = A^{\dagger}b$.

Proof. For $x_{k+1} = x_k - \gamma_k A^*(Ax_k - b)$, we compute

$$\begin{aligned} q(x_{k+1}) &= \frac{1}{2}\|Ax_{k+1} - b\|^2 \\ &= \frac{1}{2}\left(\langle Ax_k - b, Ax_k - b\rangle - 2\langle Ax_k - b, \gamma_k Ag_k\rangle + \langle \gamma_k Ag_k, \gamma_k Ag_k\rangle\right) \\ &= q(x_k) - \gamma_k\|g_k\|^2 + \frac{1}{2}\gamma_k^2\|Ag_k\|^2 \\ &= q(x_k) - \gamma_k\|g_k\|^2\left(1 - \frac{1}{2}\gamma_k\frac{\|Ag_k\|^2}{\|g_k\|^2}\right). \end{aligned} \tag{3.7}$$

Based on the steps 4 and 5 from Algorithm 3.1 immediately follows

$$\gamma_k \leq 2(1 - \varepsilon)\frac{\|g_k\|^2}{\|Ag_k\|^2},\ k \geq 1,$$

and one can simply verify

$$1 - \frac{1}{2}\gamma_k\frac{\|Ag_k\|^2}{\|g_k\|^2} \geq \varepsilon > 0, \quad k \geq 1. \tag{3.8}$$

Therefore, it is clear that the functional q is strictly monotone decreasing and it is also bounded below with zero. Thus, it follows that the sequence $q(x_k)$ converges to its minimum.

Taking into account that

$$\begin{aligned} 0 &= \lim_{k\to\infty} \mid q(x_{k+1}) - q(x_k) \mid \\ &= \lim_{k\to\infty} \gamma_k\|g_k\|^2\left(1 - \frac{1}{2}\gamma_k\frac{\|Ag_k\|^2}{\|g_k\|^2}\right), \end{aligned}$$

as well as the fact that the sequences (γ_k) and $\left(1 - \frac{1}{2}\gamma_k\frac{\|Ag_k\|^2}{\|g_k\|^2}\right)$ are bounded by real constants, one can conclude that

$$\lim_{k\to\infty} \|g_k\|^2 = 0.$$

Now, from $\lim_{k\to\infty} \|g_k\| = 0$ we have

$$\lim_{k\to\infty} A^*Ax_k = A^*b.$$

Using known fact that u is a least-squares solution of $Ax = b$ if and only if u is a solution of the "normal" equation $A^*Ax_k = A^*b$ (see [19]), we conclude that the sequence x_k converges to a least-squares solution of the operator equation $Ax = b$.

For the second part of the proof see the convergence theorem for steepest descent from [22]. □

4 SC method for computing matrix pseudoinverses

Our motivation in this section is the following result from [4] which establishes a relation between the $\{1,3\}$-inverses and the least-squares solutions of $Ax = b$.

Proposition 4.1. [4] Let $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$. The norm $\|Ax - b\|$ is smallest when $x = A^{(1,3)}b$, where $A^{(1,3)} \in A\{1,3\}$. Conversely, if $X \in \mathbb{C}^{n\times m}$ has the property that $\|Ax - b\|$ is smallest for all b, when $x = Xb$, then $X \in A\{1,3\}$.

Furthermore, our additional motivation is the next result (see, for example [4]): $X \in A\{1,3\}$ if and only if X is a least-squares solution of $AX = I_m$, i.e., X minimizes the norm $\|AX - I\|_F$.

Let $\mathcal{H} = \mathbb{C}^{n\times m}$ and $\mathcal{K} = \mathbb{C}^{m\times m}$ be regarded as Hilbert spaces. Here, on the space of complex matrices we consider the Frobenius scalar product, $\langle A, B\rangle = Tr(A^*B)$, and the Frobenious norm $\|A\|_F = \sqrt{\langle A, A\rangle}$, where $\mathrm{Tr}(A)$ denotes the trace of the matrix A. Any matrix $A \in \mathbb{C}^{m\times n}$ defines a mapping from $\mathcal{H}$ to $\mathcal{K}$ by $A(X) = AX$.

In this way, we can establish an analogy to the results of the previous sections regarding the functional $Q(X) = \frac{1}{2}\|AX - I\|_F^2$. Consequently, we obtain the iterations

$$X_{k+1} = X_k - \gamma_k G_k = X_k - \gamma_k A^*(AX_k - I), \quad k \geq 0, \tag{4.1}$$

which are of the form (3.2) and the corresponding stepsizes for SC method given by

$$\gamma_{k+1}^{SC} = \begin{cases} \frac{\langle S_k, R_k\rangle}{\langle Y_k, R_k\rangle}, & \langle Y_k, R_k\rangle > 0 \\ \frac{\|S_k\|}{\|Y_k\|}, & \langle Y_k, R_k\rangle \leq 0 \end{cases}, \quad k \geq 0, \tag{4.2}$$

where $S_k = X_{k+1} - X_k$, $Y_k = G_{k+1} - G_k$ and $R_k = S_k - \gamma_k Y_k$.

Finally, we define the following algorithm for computing $\{1,3\}$-inverses (and particularly the Moore-Penrose inverse) of complex matrices.

Algorithm 4.1 SC method for computing $\{1,3\}$-inverses of a matrix

```
Require: Complex matrix $A \in \mathbb{C}^{m\times n}$, initial approximation matrix $X_0 \in \mathbb{C}^{n\times m}$ and real positive
    constants $0 < \varepsilon \ll 1, \quad 0 < \xi_1 \ll \frac{2(1-\varepsilon)}{\|A\|^2}$.
1: Set $k = 0$, compute $Q(X_0)$, $G_0$ and use $\gamma_0 = 1$.
2: If test criteria are fulfilled then go to Step 7; otherwise, go to the next step.
3: Compute $X_{k+1}$ using (4.1), $Q(X_{k+1})$, $G_{k+1}$, $S_k = X_{k+1} - X_k$, $Y_k = G_{k+1} - G_k$.
4: Determine $\xi_2^{(k+1)} = 2(1-\varepsilon)\frac{\|G_{k+1}\|^2}{\|AG_{k+1}\|^2}$.
5: Compute the stepsize $\gamma_{k+1}$ using (4.2). If $\gamma_{k+1} < \xi_1$ or $\gamma_{k+1} > \xi_2^{(k+1)}$, set $\gamma_{k+1} = \xi_2^{(k+1)}$.
6: Set $k := k + 1$ and go to Step 2.
7: Return $X_{k+1}$ and $Q(X_{k+1})$.
```

It is clear that the iterations (4.1) can be considered as a general gradient method for computing $\{1,3\}$-inverses of a given matrix. Taking into account the equation (3.4), we can also consider BB method as a kind of a two-point stepsize gradient method and additionally we observe the steepest descent method which is a gradient descent method. The stepsizes for these methods are computed according to the following formulae respectively

$$\gamma_{k+1}^{BB} = \frac{\langle Y_k, S_k\rangle}{\langle Y_k, Y_k\rangle} \quad \text{and} \quad \gamma_k^{SD} = \frac{\|G_k\|^2}{\|AG_k\|^2} \quad k \geq 0. \tag{4.3}$$

It is known that BB method for any dimensional quadratic function is R-linearly convergent [8] as well as that the steepest descent method converges to a least-squares solution of the matrix equation $AX = I$ [22].

The algorithms (BB and steepest descent) for computing $\{1,3\}$-inverses and the Moore-Penrose inverse of a matrix would be almost the same as Algorithm 4.1. The only difference is that SC method is implemented using the restrictions imposed in Step 5 of the algorithm on the parameter γ_k, while BB and the steepest descent methods do not make use of these restrictions. Let us indicate to a significant difference between the Algorithm 4.1 and the corresponding BB method for the pseudoinverse computation. The BB method considered here is actually nonmonotone gradient method (the positiveness of the stepsize is not mandatory). On the other hand, the SC method is a strictly monotone gradient descent method (similarly as the steepest descent method).

5 Convergence properties of gradient methods

The following proposition gives a characterization of least-squares solutions of the operator equation $Ax = b$ obtained by an arbitrary gradient method given by (3.2) which converges to the minimum of the functional defined by (3.1).

Proposition 5.1. [22] *Let $\mathcal{H}$ and $\mathcal{K}$ be given Hilbert spaces and $A \in \mathcal{L}(\mathcal{H},\mathcal{K})$ is chosen operator such that $\mathcal{R}(A)$ is closed. Let the iterative process defined by* (3.2) *converges to a least-squares solution of the operator equation $Ax = b$. Then the obtained least-squares solution is completely determined by an arbitrary chosen initial approximation $x_0 \in \mathcal{H}$ and has the following representation*

$$\lim_{k\to\infty} x_k = A^\dagger b + (I - A^\dagger A)x_0, \tag{5.1}$$

where $I \in \mathcal{L}(\mathcal{H})$ is the identity operator. Consequently, $x_0 \in \mathcal{R}(A^*)$ if and only if

$$\lim_{k\to\infty} x_k = A^\dagger b. \tag{5.2}$$

Similar representation of least-squares solutions in C^* algebras obtained by the process (3.2) is established in [12].

Problem 5.1. *It seems interesting to find explicit solution of the following problem: for an arbitrary chosen least-squares solution of the equation $Ax = b$, find corresponding vector $x_0 \in \mathcal{H}$ such that the limit $L(x_0)$ of the iterative process* (3.2) *is just equal to this lss. The solution of the problem is given in the rest of this section.*

Let us denote the limiting value of the iterative process (3.2) which starts with the initial vector x_0 by

$$L(x_0) \equiv \lim_{k\to\infty} x_k = A^\dagger b + (I - A^\dagger A)x_0.$$

The following auxiliary results will be used to get the answer to the stated problem.

Lemma 5.1. *Let $A \in \mathcal{L}(\mathcal{H},\mathcal{K})$ have a closed range, where $\mathcal{H},\mathcal{K}$ are Hilbert spaces. Then*

a) $A^\dagger A A^{(1,3)} = A^\dagger$.

b) *Let $b \in \mathcal{K}$. Then $\|Ax - b\|$ is smallest when $x = A^{(1,3)}b$. Conversely, if $X \in \mathcal{L}(\mathcal{K},\mathcal{H})$ has the property that for all b the norm $\|Ax - b\|$ is smallest for $x = Xb$, then $X \in A\{1,3\}$.*

Proof. A has the following matrix form:

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & 0 \end{bmatrix} : \begin{bmatrix} \mathcal{R}(A^*) \\ \mathcal{N}(A) \end{bmatrix} \to \begin{bmatrix} \mathcal{R}(A) \\ \mathcal{N}(A^*) \end{bmatrix},$$

where A_1 is invertible. Hence,

$$A^\dagger = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \quad A^{(1,3)} = \begin{bmatrix} A_1^{-1} & 0 \\ U & V \end{bmatrix},$$

where U, V are arbitrary linear and bounded.

a) An easy computation shows that $A^\dagger A A^{(1,3)} = A^\dagger$ holds.

b) Let

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \begin{bmatrix} \mathcal{R}(A^*) \\ \mathcal{N}(A) \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \in \begin{bmatrix} \mathcal{R}(A) \\ \mathcal{N}(A^*) \end{bmatrix}$$

be arbitrary elements from $\mathcal{H}$ and $\mathcal{K}$ respectively. We see that

$$\min \|Ax - b\|^2 = \min \|A_1x_1 - b_1\|^2 + \|b_2\|^2 = \|b_2\|^2$$

is attained for $x_1 = A_1^{-1}b_1$. Hence, all least-squares solutions of the equation $Ax = b$ have the form $\begin{bmatrix} A_1^{-1}b_1 \\ x_2 \end{bmatrix}$, which is the result proved in [11]. Let

$$x = A^{(1,3)}b = \begin{bmatrix} A_1^{-1}b_1 \\ Ub_1 + Vb_2 \end{bmatrix},$$

whence x is a least-squares solution. Conversely, for all b,

$$x = Xb = \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} X_1b_1 + X_2b_2 \\ X_3b_1 + X_4b_2 \end{bmatrix} = \begin{bmatrix} A_1^{-1}b_1 \\ x_2 \end{bmatrix}$$

since x is a least-squares solution, from which follows that $X_1 = A_1^{-1}$ and $X_2 = 0$, we get $X \in A\{1,3\}$.

□

Lemma 5.2. *Let $\mathcal{H},\mathcal{K},\mathcal{M}$ be Hilbert spaces. If $A \in \mathcal{L}(\mathcal{H},\mathcal{K})$ has a closed range and $F \in \mathcal{L}(\mathcal{M},\mathcal{H})$ satisfies $\mathcal{R}(F) = \mathcal{N}(A)$, then there exists some $G \in \mathcal{L}(\mathcal{H},\mathcal{M})$ such that $FG = I - A^\dagger A$.*

Proof. We keep the notations from Lemma 5.1. Since $\mathcal{R}(F) = \mathcal{N}(A)$, we conclude that F has the following form:

$$F = \begin{bmatrix} 0 \\ F_1 \end{bmatrix} : \mathcal{M} \to \begin{bmatrix} \mathcal{R}(A^*) \\ \mathcal{N}(A) \end{bmatrix},$$

where $F_1 : \mathcal{M} \to \mathcal{N}(A)$ is onto, hence it is right invertible. There exists some $G_1 : \mathcal{N}(A) \to \mathcal{M}$, such that $F_1 G_1 = I$. Now, let us consider the operator

$$G = \begin{bmatrix} 0 & G_1 \end{bmatrix} : \begin{bmatrix} \mathcal{R}(A^*) \\ \mathcal{N}(A) \end{bmatrix} \to \mathcal{M}.$$

Then

$$FG = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} = I - A^\dagger A,$$

which was our original attention. □

We firstly derive a particular solution to Problem 5.1.

Theorem 5.1. *For an arbitrary given operator $A \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ with closed range and an arbitrary chosen least-squares solution s of the operator equation $Ax = b$ the following holds*

$$s = L(s - A^\dagger b). \tag{5.3}$$

Proof. According to Lemma 5.1, part **b)**, we can characterize the set of all least-squares solutions of the operator equation $Ax = b$ by $\{A^{(1,3)}b + (I - A^{(1,3)}A)y |\ y \in \mathcal{H}\}$, where $A^{(1,3)}$ is arbitrary but fixed element. For example, it is possible to choose $A^{(1,3)} = A^\dagger$. Let $F \in \mathcal{L}(\mathcal{H}, \mathcal{H})$ be such that $\mathcal{R}(F) = \mathcal{N}(A)$ (for example we can take $F = \mathcal{I}_{\mathcal{H}}|_{\mathcal{N}(\mathcal{A})}$). Since $\mathcal{R}(I - A^\dagger A) = \mathcal{N}(A)$ from Lemma 5.2 we have that there exists an operator $G \in \mathcal{L}(\mathcal{H}, \mathcal{H})$ such that $FG = I - A^\dagger A$. Now it is clear that we can reduce the characterization set to the following one

$$\{A^\dagger b + Fy |\ y = Gz, z \in \mathcal{H}\},$$

Now we obtain

$$s = A^\dagger b + FGz = A^\dagger b + (I - A^\dagger A)z.$$

One can verify the following

$$s = A^\dagger b + (I - A^\dagger A)(s - A^\dagger b),$$

taking into account that $s = Sb$ where S is some $\{1, 3\}$ inverse of A. Therefore, it is possible to choose $z = s - A^\dagger b$, which implies

$$x_0 = s - A^\dagger b,$$

and completes the proof. □

In the next theorem we get a general solution to the stated problem.

Theorem 5.2. *For an arbitrary given linear operator $A \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ and an arbitrary least-squares solution s of the equation $Ax = b$ the following holds*

$$s = L(s - A^\dagger b + A^\dagger Ay),\ y \in \mathcal{H}. \tag{5.4}$$

Proof. Let us start with the least-squares solution s of the equation $Ax = b$, obtained by (3.2)

$$s = A^\dagger b + (I - A^\dagger A)x_0.$$

In order to find the vector x_0 in terms of the vector s we consider the following operator equation

$$Cx_0 = d, \tag{5.5}$$

where $C = I - A^\dagger A$, $d = s - A^\dagger b$. Since C is idempotent and Hermitian (thus the orthogonal projector), it follows $C = C^\dagger C$. According to Theorem 5.1 we conclude that the equation $Cx_0 = d$ has a solution. Following the general form of the least-squares solution (which is a solution) of the equation $Cx_0 = d$ (see [23]), we obtain

$$x_0 = C^\dagger d + (I - C^\dagger C)y, \quad y \in \mathcal{H}. \tag{5.6}$$

After applying the equality $A^\dagger As = A^\dagger b$ which is not difficult to check and $I - C^\dagger C = A^\dagger A$ we obtain

$$\begin{aligned} x_0 &= (I - A^\dagger A)(s - A^\dagger b) + A^\dagger Ay \\ &= s - A^\dagger b - A^\dagger As + A^\dagger b + A^\dagger Ay \\ &= s - A^\dagger b + A^\dagger Ay, \end{aligned} \tag{5.7}$$

which completes the proof. □

Remark 5.1. *Let us consider SC method, BB method and the steepest descent methods as gradient methods for computing a least-squares solution of the matrix equation $AX = I$ and the iterative methods introduced in* [12]. *For any initial approximation X_0 these methods converge to the $A^{(1,3)}$ inverse, given by $A^\dagger + (I - A^\dagger A)X_0$. Particularly, in the case $X_0 \in R(A^*)$ we have that these methods converge to $A^\dagger$.*

For this purpose, it is realistic to expect that the general iterative scheme (4.1) possesses the same convergence properties.

Corollary 5.1. *Let the matrix $X_0 \in \mathbb{C}^{n\times m}$ be any initial approximation, $A \in \mathbb{C}^{m\times n}$ be a given matrix and $I \in \mathbb{C}^{m\times m}$ be the identity matrix. If the sequence $(X_k)_k$ given by the gradient method* (4.1) *converges to a least-squares solution of the matrix equation $AX = I$, then this solution is given by*

$$\lim_{k\to\infty} X_k = A^\dagger + (I - A^\dagger A)X_0. \tag{5.8}$$

Particularly, $X_0 \in \mathcal{R}(A^*)$ if and only if

$$\lim_{k\to\infty} X_k = A^\dagger. \tag{5.9}$$

Proof. Follows straight from Proposition 5.1. □

Remark 5.2. *According to Corollary* 5.1*, we conclude that the $\{1,3\}$-inverse which is achieved by the iterative process* (4.1) *(as a least-squares solution) is completely determined by the initial approximation X_0, and it is given by $A^\dagger + (I - A^\dagger A)X_0$. At this point, we denote the limit of the iterative process* (4.1) *determined by X_0 as $L(X_0)$. Now we consider algebraic properties of the set $\{L(X_0) \mid X_0 \in \mathbb{C}^{n\times m}\}$.*

Corollary 5.2. *Let $A \in \mathbb{C}^{m\times n}$ be given complex matrix. The following statement holds*

$$L = \left\{L(X_0) \mid X_0 \in \mathbb{C}^{n\times m}\right\} = A\{1,3\}. \tag{5.10}$$

Proof. We use the following characterization of the set $A\{1,3\}$ from [4]:

$$A\{1,3\} = \left\{A^{(1,3)} + (I - A^{(1,3)}A)Z : Z \in \mathbb{C}^{n\times m}\right\},$$

for arbitrary but fixed $A^{(1,3)} \in A\{1,3\}$. For example, it is possible to use $A^{(1,3)} = A^\dagger$. In this case, the inclusion $L \subseteq A\{1,3\}$ is evident. To verify the opposite inclusion let us choose an arbitrary $A^{(1,3)} \in A\{1,3\}$. It is of the form $A^{(1,3)} = A^\dagger + (I - A^\dagger A)Z$, $Z \in \mathbb{C}^{n\times m}$. If the initial iteration in (4.1) is chosen as $X_0 = Z$ we obtain $A^{(1,3)} = L(X_0)$, which implies $A\{1,3\} \subseteq L$. □

Let us consider the analogous problem with respect to Problem 5.1 considering the matrix equation $AX = I$; for an arbitrary chosen $\{1, 3\}$-inverse $A^{(1,3)}$ find corresponding matrix X_0 such that the limit $L(X_0)$ of the iterative process (5.8) is just equal to $A^{(1,3)}$.

In the following corollary we present the general solution of the stated problem.

Corollary 5.3. *For an arbitrary given matrix $A \in \mathbb{C}^{m\times n}$ and an arbitrary chosen $S \in A\{1, 3\}$ the following holds*

$$S = L(S - A^\dagger + A^\dagger AY),\ Y \in \mathbb{C}^{n\times m}. \tag{5.11}$$

It is possible to derive an alternative characterization for the convergence of (4.1) using main principle from [24]. If L is the desired limit matrix and X_k is the k-th estimate of L, then the convergence properties of the examined algorithm can be studied with the aid of the error matrix $E_k = X_k - L$. If an iterative algorithm is expressible as a simple matrix formula, E_{k+1} is a sum of several terms:

- zero-order term consisting of a matrix which does not depend upon E_k,
- one or more first-order matrix terms in which E_k or its conjugate transpose E_k^* appears only once,
- higher-order terms in which E_k or E_k^* appears at least twice.

All suitable algorithms have a zero-order term equal to 0. Hence the first-order terms determine the terminal convergence properties [24]. The calculation of the first-order terms $error_1$ begins by substituting $X_k = A^\dagger + E$ and expanding the resulting formula. Using this approach, in the following statement we verify the linear convergence of our method (4.1).

Theorem 5.3. *Iterative method* (4.1) *converges to the Moore-Penrose inverse $X = A^\dagger$ linearly, where the first-order and the second-order terms, corresponding to the error estimation of* (4.1) *are equal to:*

$$error_1 = (I - \gamma_k A^* A)E_k, \quad error_2 \ = 0, \tag{5.12}$$

respectively.

Proof. Putting $X_k = A^\dagger + E_k$ in (4.1) it is not difficult to verify that the error matrix E_{k+1} is equal to

$$E_{k+1} = E_k - \gamma_k A^* AA^\dagger - \gamma_k A^* AE_k + \gamma_k A^*,$$

which confirms the statements in (5.12). □

6 Numerical Results

According to the convergence properties of gradient methods (which include the steepest descent method, SC and BB method), investigated in Section 5, it seems reasonable to compare these methods in computation of the ordinary inverse, the Moore-Penrose inverse and various $\{1, 3\}$-inverses. The code for the three methods (the steepest descent, BB and SC method) is written in the MATLAB programming package and tested on a Workstation Intel Core duo 1.6 GHz. We selected 5 different types of matrices as test problems. For each test matrix we have considered five different dimensions which are chosen according to the nature of the test problem. For each test problem we compared two indicators: number of iterations and the accuracy of the obtained result, i.e., the difference between the exact pseudoinverse and its approximation (obtained by the algorithm) given in terms of the matrix norm. Stopping conditions are:

$$\|X_{k+1} - X_k\|_F \le \varepsilon = 10^{-8} \ \text{ and } \ |f_{k+1} - f_k| \le \varepsilon = 10^{-8}.$$

Example 6.1. *In this example the inverse of the nonsingular symmetric matrix Z_n of order $n = 2, 4, 6, \ldots$, taken from* [30]*, which is given by*

$$z_{i,j} = \begin{cases} a - 1, & i = j,\ i \text{ even} \\ a + 1, & i = j,\ i \text{ odd} \\ a, & \text{otherwise} \end{cases} \tag{6.1}$$

is computed. For computing the ordinary inverse of the parametric matrix Z_n we use $a = 2$.

Table 6.1. Numerical results for computing the inverse of the matrix Z_n where $a = 2$

	Number of iterations			$\|Z^{-1} - X\|_F$		
Dim	SC	BB	Steepest	SC	BB	Steepest
10	5	8	5	1.4e-11	6.9e-07	2.9e-09
20	5	13	5	4.6e-10	1.9e-09	3.4e-06
30	5	13	5	3.2e-09	6.6e-09	3.4e-05
40	5	13	5	6.6e-09	9.2e-08	2.7e-04
50	7	8	5	3.9e-08	3.5e-05	7.9e-04

Following the results from Table 6.1 it is evident that SC method as well as the steepest descent method outperform BB method approximately twice, in the number of iterations. Also, according to the accuracy, the SC method proves oneself as the best. It is important to say that the matrix Z_n is well conditioned matrix and therefore the steepest descent method is competitive with the two-point stepsize methods.

Example 6.2. *The structured (Toeplitz) test matrix*

$$A_n = \text{toeplitz}[(1, 1/2, \ldots, 1/(n-1), 1/n)]$$

is taken from [3] *and the numerical results for computing its inverse are presented.*

Table 6.2. Numerical results for computing the inverse of the Toeplitz matrix A_n

	Number of iterations			$\|A^{-1} - X\|_F$		
Dim	SC	BB	Steepest	SC	BB	Steepest
10	83	72	618	4.6e-07	2.2e-08	2.8e-07
20	117	108	903	2.4e-07	1.9e-08	5.1e-07
30	124	127	1237	4.1e-07	1.4e-07	7.1e-07
40	150	137	1533	5.6e-07	1.4e-06	8.5e-07
50	168	162	1789	6.9e-07	6.5e-07	9.8e-07

From the results for the computation of the inverse of the well conditioned matrix A_n we conclude that the steepest descent method is not competitive with the two-point stepsize methods regarding the number of iterations. Additionally, the BB method performs slightly better then SC method. Also, there is no big difference between the accuracy for all three observed methods.

Example 6.3. *In this example we consider the symmetric test matrix S_n of order $n = 3, 5, 7, \ldots$ and of rank $n-1$, taken from* [30]*, which is given by*

$$s_{i,j} = \begin{cases} a-1, & i = j,\ i \text{ even} \\ a+1, & i = j,\ i \text{ odd} \\ a+1, & |i-j| = n-1 \\ a, & \text{otherwise.} \end{cases} \tag{6.2}$$

For this ill-conditioned matrix whose condition number is large $cond_F(S_n) \approx |a|^2(n^2 - 3n/2)$, $|a| \gg 2$, the Moore-Penrose inverse is determined. The presented numerical results are obtained after we made the choice $a = 2$.

Table 6.3. Numerical results for computing the Moore-Penrose inverse of the matrix S_n

	Number of iterations			$\|S^\dagger - X\|_F$		
Dim	SC	BB	Steepest	SC	BB	Steepest
9	12	17	216941	1.6e-04	1.8e-06	2.6e-04
11	12	17	315545	4.6e-05	2.9e-06	4.9e-04
13	9	17	449687	1.9e-05	3.0e-05	8.7e-04
15	9	17	629469	1.0e-05	1.2e-04	0.0014
17	9	17	863949	6.3e-06	4.1e-04	0.0022

According to the results from Table 6.3 it is clear that two-point stepsize methods behave significantly better with respect to the steepest descent method not only in the number of iterations but also in the accuracy The enormous number of iterations corresponding to the steepest descent method confirms the fact that the steepest descent is very badly affected by ill conditioning. Additionally, SC method outperform BB method observing the number of iterations almost as twice.

Example 6.4. *The tridiagonal square test matrix B_n of the order n and* $\operatorname{rank}(B_n) = n - 1$ *taken from* [30]

$$B_n = \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & \mathbf{0} & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & \mathbf{0} & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}$$

is considered. The numerical results for computing the Moore-Penrose inverse $B_n^\dagger$ as well as the $B_n^{(1,3)}$ inverse are presented. The matrix B_n is ill-conditioned matrix with the spectral condition number equal to $cond_2(B_n) = 4n^2/\pi^2$.

Table 6.4. Numerical results for computing the Moore-Penrose inverse of the matrix B_n

	Number of iterations			$\|B^\dagger - X\|_F$		
Dim	SC	BB	Steepest	SC	BB	Steepest
5	71	42	585	7.7e-07	4.6e-07	3.3e-07
10	340	170	11369	5.8e-06	1.8e-07	5.8e-06
15	924	671	55215	1.7e-05	1.5e-05	2.9e-05
20	2130	2058	168083	2.3e-04	8.6e-07	9.5e-05
30	3125	4669	801031	7.4e-04	2.1e-07	4.8e-04

To compute the $B_n^{(1,3)}$ inverse we made the choice $X_0 = I$ for the initial approximation. Each of the observed methods converges to $B^\dagger + I - B^\dagger B$, which is completely determined by $X_0 = I$.

Table 6.5. Numerical results for computing $B^{(1,3)}$ inverse of the matrix B_n

	Number of iterations			$\|B^{(1,3)} - X\|_F$		
Dim	SC	BB	Steepest	SC	BB	Steepest
5	63	44	629	6.3e-08	3.5e-10	3.3e-07
10	365	413	11253	2.2e-06	1.1e-05	5.8e-06
15	847	517	54843	6.7e-05	3.4e-06	3.0e-05
20	2254	1813	167147	2.5e-04	2.3e-04	9.6e-05
30	4096	3716	797241	1.0e-03	1.3e-03	4.9e-04

SC and BB method show significantly better results with respect to the steepest descent method for computing the Moore-Penrose inverse as well as the $\{1,3\}$ inverse of the matrix B_n, as it is excepted. Although BB method outperform SC method observing the number of iterations, in some cases such as the case $n = 30$ from Table 6.4 we see a big difference in the number of iterations in favor of SC method.

Example 6.5. *The test matrix A_n is constructed by the Matrix Market generator. The matrix A_n is nonsymmetric square sparse* (40% *zero elements*) *random matrix which is filled out by uniformly distributed elements from the interval* $[0,5]$.

Table 6.6. Numerical results for computing the Moore-Penrose inverse of the matrix A_n

	Number of iterations			$\|A^{\dagger} - X\|_F$		
Dim	SC	BB	Steepest	SC	BB	Steepest
5	57	42	253	4.5e-07	1.5e-07	2.6e-07
10	478	485	23763	3.7e-05	1.5e-06	2.4e-05
15	445	419	13567	8.9e-06	3.0e-05	1.2e-05
20	384	300	8955	1.7e-05	1.4e-05	9.0e-06
30	1532	1305	137873	2.0e-04	1.4e-04	1.7e-04

For the random sparse matrix A_n, according to presented numerical results, again we see the ascendancy of the two point stepsize methods over the steepest descent method. Also, we distinguish the slightly better performance of BB method with respect to SC method.

7 Conclusion

The usage of the quasi-Newton methods for finding an optimal solution of a given function is a natural continuation of the idea that can be described as an application of the nonlinear optimization in computing least-squares solutions of the operator equation $Ax = b$. This strategy adapted for functionals on Hilbert spaces can be also successfully used for computing $\{1,3\}$-inverses, including the Moore-Penrose inverse, of a given complex matrix.

In this paper we presented the previously mentioned adaptation, i.e. we did an extension of the secant equation to Hilbert spaces. Furthermore, using the idea of the scalar correction method (SC method) we established a new gradient method for computing least-squares solutions of the equation $Ax = b$ on Hilbert spaces, which in addition to good properties of the secant equation, also preserves monotonicity.

The set of least-squares solutions of the operator equation $Ax = b$ is identified with the set of limits of the iterative process (3.2). Any limit value of the iterative process given by (3.2) is completely described by the initial approximation, and it is a least-squares solution. Conversely for a given least-squares solution we found the set of all initial approximations for the iterative process which lead to that least-squares solution. As should be expected, respective results are presented regarding the computation of $\{1,3\}$-inverses as well as the Moore-Penrose inverse of a given complex matrix $A \in \mathbb{C}^{m\times n}$.

The presented numerical results confirm the expectation relative to the bad convergence properties of the steepest descent method for ill-conditioned problems. And thus make favorable two point stepsize methods with respect to the steepest descent method. Additionally, regarding the numerical results we conclude that the scalar correction method is competitive with the favorable Barzilai Borwein method, not only in the number of iterations but also in the accuracy. Also, the complexity of the SC algorithm is very similar to the complexity of the BB method, which is known as a method easy for the implementation.

References

[1] H. Akaike, *On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method*, Ann. Inst. Statist. Math. Tokyo **11** (1959), 1–17.

[2] J. Barzilai and J.M. Borwein. *Two point step size gradient method*, IMA J. Numer. Anal. **8** (1988), 141–148.

[3] D. Bini and B. Meini, *Approximate displacement rank and applications*, in Structured Matrices in Mathematics, Computer Science and Engineering II, V. Olshevsky Editor, Contemporary Mathematics **281** (2001), 215–232, American Mathematical Society, Rhode Island.

[4] A. Ben-Israel, T.N.E. Greville, Generalized Inverses: Theory and Applications, second ed., Springer, 2003.

[5] C. Brezinski, *A classification of quasi-Newton methods*, Numer. Algor. **33** (2003), 123–135.

[6] J. F. Cai, M. K. Ng, Y. Wei, *Modified Newton's Algorithm for Computing the Group Inverses of Singular Toeplitz Matrices*, J. Comput. Math. **24** (2006), 647–656.

[7] L. Collatz, *Functional analysis and numerical mathematics*, Academic Press, New York, 1966.

[8] Y.H. Dai, L.Z. Liao, *R-linear convergence of the Barzilai and Borwein gradient method*, IMA J. Numer. Anal. **22** (2002), 1–10.

[9] Y. H. Dai, H. Zhang, *Adaptive two-point stepsize gradient algorithm*, Numer. Algor. **27** (2001), 377–385.

[10] Y.H. Dai, *On the Nonmonotone Line Search*, J. Optim. Theory Appl. **112** (2002), 315–330.

[11] D. S. Djordjevic and V. Rakocevic, *Lectures on generalized inverses*, Faculty of Sciences and Mathematics, University of Niš, 2008.

[12] D.S. Djordjevic, P.S. Stanimirović, *Iterative methods for computing generalized inverses related with optimization methods*, J. Aust. Math. Soc. 78 (2005), 257–272.

[13] R. Fletcher, *On the Barzilai-Borwein method*, Numerical Analysis Report NA/207, (2001).

[14] G.E. Forsythe, *On the asimptotic directions of the s-dimensional optimum gradient method*, Numer. Math. **11** (1968), 57–76.

[15] M.R.Hestenes, *Pseudoinverses and conjugate gradients*, Commun. ACM **18** (1975), 40–43.

[16] W.J. Kammerer, M.Z. Nashed, *On the convergence of the conjugate gradient method for singular linear operators equations*, SIAM J. Numer. Anal. **97** (1972), 165–181.

[17] L.V. Kantorovich, G. P. Akilov, Functional analysis (in Russian), Moscow, 1977.

[18] J. E. Marsden, T. Radiu, *Manifolds, Tensor Analysis, and Application*, Springer-Verlag New York, Inc, 2001.

[19] S.G. Mikhlin, *The Problem of the Minimum of a Quadratic Functional*, Holden-Day, San Francisco, 1965.

[20] M. Miladinović, P. Stanimirović, S. Miljković, *Scalar Correction Method for Solving Large Scale Unconstrained Minimization Problems*, J. Optim. Theory. Appl. **151** (2011), 304–320.

[21] A. Mohsen, J. Stoer, *A Variable Metric Method for Approximating Generalized Inverses of Matrices*, ZAMM - Z. Angew. Math. Mech. **81** (2001), 435446.

[22] M.Z. Nashed, *Steepest descent for singular linear operators equations*, SIAM J. Numer. Anal. **7** (1970), 358–362.

[23] M.Z. Nashed, *Generalized Inverses and Applications, Edited by M. Zuhair Nashed*, Proceedings of an Advanced Seminar, Academic Press, New York, San Francisko, London, 1976.

[24] W. H. Pierce, *A Self-Correcting Matrix Iteration for the Moore-Penrose Generalized Inverse*, Linear Algebra Appl. **244** (1996), 357–363.

[25] M. Raydan, *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA J. Numer. Anal. **13** (1993), 321–326.

[26] M. Raydan, B.F. Svaiter, *Relaxed Steepest Descent and Cauchy-Barzilai-Borwein Method*, Comput. Optim. Appl. **21** (2001), 155–167.

[27] P.S. Stanimirović, M. Miladinović, *Accelerated gradient descent methods with line search*, Numer. Algor. **54** (2010), 503–520 .

[28] K. Tanabe, *Conjugate-gradient method for computing the Moore-Penrose inverse and rank of a matrix*, J. Optim. Theory. Appl. **22** (1977), 1-23.

[29] Y. Wei, J. Cai, M. K. Ng, *Computing Moore-Penrose Inverses of Toeplitz Matrices by Newton's Iteration*, Math. Comput. Modelling, **40** (2004), 181–191.

[30] G. Zielke, *Report on test matrices for generalized inverses*, Computing **36** (1986), 105–162.