





Article

Improving Newton–Schulz Method for Approximating Matrix Generalized Inverse by Using Schemes with Memory

Alicia Cordero ^{1,*}, Javier G. Maimó ², Juan R. Torregrosa ¹ and María P. Vassileva ²

¹ Institute for Multidisciplinary Mathematics, Universitat Politècnica de València, 46022 Valencia, Spain; jrtorre@mat.upv.es

² Área de Ciencias Básicas y Ambientales, Instituto Tecnológico de Santo Domingo (INTEC), Av. Los Procéres, Gala, Santo Domingo 10602, Dominican Republic; javier.garcia@intec.edu.do (J.G.M.); maria.penkova@intec.edu.do (M.P.V.)

* Correspondence: acordero@mat.upv.es

Abstract: Some iterative schemes with memory were designed for approximating the inverse of a nonsingular square complex matrix and the Moore–Penrose inverse of a singular square matrix or an arbitrary $m \times n$ complex matrix. A Kurchatov-type scheme and Steffensen’s method with memory were developed for estimating these types of inverses, improving, in the second case, the order of convergence of the Newton–Schulz scheme. The convergence and its order were studied in the four cases, and their stability was checked as discrete dynamical systems. With large matrices, some numerical examples are presented to confirm the theoretical results and to compare the results obtained with the proposed methods with those provided by other known ones.

Keywords: nonlinear matrix equations; inverse and pseudo-inverse matrices; iterative procedure; methods with memory

MSC: 65H99; 15A09



Citation: Cordero, A.; Maimó, J.G.; Torregrosa, J.R.; Vassileva, M.P. Improving Newton–Schulz Method for Approximating Matrix Generalized Inverse by Using Schemes with Memory. *Mathematics* **2023**, *11*, 3161. <https://doi.org/10.3390/math11143161>

Academic Editor: Changbum Chun

Received: 29 May 2023

Revised: 13 July 2023

Accepted: 14 July 2023

Published: 18 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Calculating the inverse of a matrix, especially of a large size, is a very difficult task with a high computational cost. The alternative is the use of iterative algorithms to estimate it. In a vast range of fields, such as image and signal processing [1–4], encryption [5,6], control system analysis [7,8], etc., it is necessary to calculate the inverse or different generalized inverses in order to solve the problems posed.

In recent years, many iterative schemes of different orders of convergence have been designed to estimate the inverse of a complex matrix A or some generalized inverse (Moore–Penrose inverse, Drazin inverse, etc.). In 2013, Weiguo et al. in [9] constructed a sequence of third-order iterations converging to the Moore–Penrose matrix. In the same year as well, Toutounian and Soleymani in [10] presented a high-order method for approximating inverses and pseudo-inverses of complex matrices based on Homeier’s scheme, with a derivative-free composition. More recently, Stanimirovic et al. in [11] designed efficient transformations of the hyperpower iterative method for computing the generalized inverses $A_{T,S}^{(2)}$ with the aim of minimizing the number of required matrices by matrix products per cycle. In 2020, Kaur et al. established in [12] new formulations of the fifth-order hyperpower method to compute the weighted Moore–Penrose inverse, improving the efficiency indices. Such approximations were found to be robust and effective when implemented as a preconditioned matrix to solve the linear systems. All these schemes were designed with the starting point of iterative procedures without memory, that is where each new iteration is calculated using only the information provided by the previous iterate.

Iterative procedures that use more than one previous iterate to calculate the next one are called methods with memory. In the context of matrix inverse approximations, the secant scheme was proposed by the authors in [13]. For a nonsingular matrix, the secant method gives an estimation of the inverse and, when the matrix is singular, an approximation of the pseudo-inverse and Drazin inverse. Furthermore, the superlinear convergence was demonstrated in all cases.

In this manuscript, we focused on constructing several iterative methods with memory, free of inverse operators and with different orders of convergence, for finding the inverse of a nonsingular complex matrix. We also analyzed the proposed schemes for computing the Moore–Penrose inverse of complex rectangular matrices. On the other hand, these procedures allow approximating other generalized inverses such as the Drazin inverse, the group inverse, etc., although this is beyond the scope of this work.

Let A be a complex $n \times n$ nonsingular matrix. The design of iterative algorithms without memory for estimating the inverse matrix, which we call Newton–Schulz-type methods, is mostly based on iterative solvers for the scalar equation $f(x) = 0$ applied to the nonlinear matrix equation:

$$F(X) = X^{-1} - A = 0, \tag{1}$$

where $F : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$ is a nonlinear matrix function.

The most-used iterative scheme to approximate A^{-1} is the Newton–Schulz scheme [14]:

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

where I is an identity matrix of size $n \times n$.

On the other hand, in the context of iterative procedures with memory, the authors presented in [13] a secant-type method (SM), whose iterative expression is:

$$X_{k+1} = X_{k-1} + X_k - X_{k-1}AX_k, \quad k = 0, 1, 2, \dots, \tag{3}$$

where X_0, X_{-1} are the starting guesses. With a particular choice of the initial approximations, the authors proved that sequence $\{X_k\}$, obtained by (3), converges to A^{-1} with order of convergence 1.618.

To analyze the convergence order of the iterative methods with memory for solving nonlinear equations $f(x) = 0$, the R -order is used (see [15]), which we summarize below.

Theorem 1. *Let ψ be an iterative method with memory that generates a sequence $\{x_k\}$ of approximations to the root α of $f(x) = 0$, and let this sequence converge to α . If there exists a nonzero constant η and nonnegative numbers $t_i, i = 0, 1, \dots, m$, such that the inequality:*

$$|e_{k+1}| \leq \eta \prod_{i=0}^m |e_{k-i}|^{t_i}$$

holds, then the R -order of convergence of the iterative method ψ satisfies the inequality:

$$O_R(\psi, \alpha) \geq s^*,$$

where s^* is the unique positive root of the equation:

$$s^{m+1} - \sum_{i=0}^m t_i s^{m-i} = 0.$$

Here, $e_k = x_k - \alpha$ denotes the error of the approximation in the k th iterative step.

Continuing with the idea of designing iterative procedures with memory, in this work, we used the scalar iterative methods of Kurchatov and Steffensen with memory, which we adapted to the context of matrix equations. In the scalar case, this kind of procedure can

reach a higher order of convergence than schemes without memory with the same number of functional evaluations and usually has better stability properties. However, as far as we know, only the secant scheme has been extended to matrix equations with good results, in [13].

Kurchatov’s scheme is an iterative algorithm with memory with quadratic convergence to solve scalar equations $f(x) = 0$ (see [16]), which is deduced from Newton’s method by substituting $f'(x)$ by Kurchatov’s divided difference, that is:

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, x_{k-1}]_K} = x_k - \frac{2(x_k - x_{k-1})f(x_k)}{f(2x_k - x_{k-1}) - f(x_{k-1})}, k = 1, 2, \dots \tag{4}$$

Regarding Steffensen’s scheme, it was initially developed in [17], where the derivative $f'(x)$ was replaced by the divided difference $f[x_k, x_k + f(x_k)]$. It still holds the original quadratic convergence, but in practice, the set of converging initial estimations is quite smaller than that of Newton’s scheme. It was Traub who, in [18], introduced an accelerating parameter γ in the divided difference such that $f'(x_k) \approx f[x_k, x_k + \gamma f(x_k)]$ derived an iterative method whose error equation was:

$$e_{k+1} = (1 + \gamma f'(\alpha))c_2 e_k^2 + O(e_k^3),$$

with $c_2 = \frac{1}{2} \frac{f''(\alpha)}{f'(\alpha)}$. With the aim of increasing the order of convergence of the scheme, Traub defined an approximation of $\gamma = -\frac{1}{f'(\alpha)}$ by $\gamma_k = -\frac{1}{f[x_k, x_{k-1}]}$, obtaining the procedure:

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k + \gamma_k f(x_k), x_k]}, k = 0, 1, 2, \dots,$$

which, given initial values x_0 and γ_0 , has order of convergence $1 + \sqrt{2}$.

The stability of these schemes with memory for scalar problems was analyzed firstly in [19,20], showing very stable performance in both cases. In what follows, some definitions and properties about vectorial discrete dynamical systems are introduced. These tools are shown to be useful in the following sections.

Basic Concepts of Qualitative Studies of Schemes with Memory

Let us start with an iterative scheme with memory, which is used to solve the scalar problem $f(x) = 0$, using two previous iterates to calculate the next one:

$$x_{k+1} = \psi(x_{k-1}, x_k), k \geq 1,$$

with x_0 and x_1 being its seeds. Therefore, a solution α is estimated whether $x_{k+1} = x_k$ or, equivalently, $x_k = \psi(x_{k-1}, x_k)$. This estimation can be obtained as a fixed point of $\Psi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ by means of:

$$\Psi(x_{k-1}, x_k) = (x_k, \psi(x_{k-1}, x_k)), k = 1, 2, \dots,$$

with again x_0 and x_1 being the seeds.

To study the qualitative behavior of a fixed point iterative scheme with memory, it is applied on a low-degree polynomial $p(x)$, generating a vectorial rational operator, $G(x)$. Then, the orbit of $(z, x) \in \mathbb{R}^2$ is the set:

$$\{(z, x), G((z, x)), \dots, G^m((z, x)), \dots\}.$$

Therefore, a point (z, x) is fixed if it is the only point belonging to its orbit, and it is a T-periodic point if its orbit is only defined by $\{(z, x), G((z, x)), \dots, G^{T-1}((z, x))\}$ and $G^T((z, x)) = (z, x)$.

On the other hand, the orbit of a point can be classified depending on its asymptotic behavior, according to the next result.

Theorem 2 ([21], p. 558). Let $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be C^2 . Let us also assume that x^* is a period- k point. Let $\lambda_i, i = 1, 2, \dots, n$ be the eigenvalues of the Jacobian $G'(x^*)$:

- (a) If $|\lambda_j| < 1, j = 1, 2, \dots, n$, then x^* is an attractor. It is said to be a superattractor if $|\lambda_j| = 0$.
- (b) If λ_{j_0} exists such that $|\lambda_{j_0}| > 1$, then the periodic point is unstable (repelling or saddle).
- (c) If $|\lambda_j| > 1, j = 1, 2, \dots, n$, then x^* is a repulsor.

Moreover, we define a point $x_c \in \mathbb{R}^2$ as a critical point of G if $\det(G'(x_c)) = 0$. Indeed, if x_c is not a zero of $p(x)$, it is named a free critical point. In a similar way, if a fixed point is not a zero of $p(x)$, it is named a strange fixed point.

Denoting by $x^F \in \mathbb{R}^2$ an attracting fixed point of: G , its basin of attraction $\mathcal{B}(x^F)$ is defined as:

$$\mathcal{B}(x^F) = \left\{ (z, x) \in \mathbb{R}^n : G^m((z, x)) \rightarrow x^F, m \rightarrow \infty \right\}.$$

The union of all the basins of attraction of G is defined as the Fatou set, $\mathcal{F}(G)$, and its complementary set in \mathbb{R}^2 is the Julia set $\mathcal{J}(G)$. It holds all the repelling fixed points and sets the boundary among the basins of attraction.

In Sections 2 and 3, we build a Kurchatov-type and a Steffensen-type iterative method with memory, respectively, to estimate the inverse of a nonsingular complex matrix. Both schemes are free of inverse operators, and we prove their order of convergence and stability. In Section 4, we extend these schemes to calculate the Moore–Penrose inverse of a complex rectangular matrix. Section 5 is devoted to the numerical tests to analyze their performance and confirm the theoretical results. We end the work with some conclusions.

2. Kurchatov-Type Method

The Kurchatov divided difference was defined by [16] as:

$$f[x_{k-1}, 2x_k - x_{k-1}; f]_K = \frac{f(2x_k - x_{k-1}) - f(x_{k-1})}{2(x_k - x_{k-1})}.$$

For an equation $f(x) = 0$, Kurchatov’s method has order of convergence two and iterative expression:

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, x_{k-1}]_K} = x_k - \frac{2f(x_k)(x_k - x_{k-1})}{f(2x_k - x_{k-1}) - f(x_{k-1})}.$$

If we apply this method to $F(X) = X^{-1} - A$, where $F : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$ and $A \in \mathbb{C}^{n \times n}$ is an $n \times n$ nonsingular matrix, not necessarily diagonalizable, we obtain:

$$\begin{aligned} X_{k+1} &= X_k - 2[F(2X_k - X_{k-1}) - F(X_{k-1})]^{-1}F(X_k)(X_k - X_{k-1}) \\ &= X_k - 2[(2X_k - X_{k-1})^{-1} - X_{k-1}^{-1}]^{-1}(X_k^{-1} - A)(X_k - X_{k-1}). \end{aligned} \tag{5}$$

Next, we show a technical lemma whose result we use later.

Lemma 1. Let $A, B \in \mathbb{C}^{n \times n}$ be invertible and commutative, $AB = BA$, then:

$$(2A - B)^{-1} - B^{-1} = 2(2A - B)^{-1}B^{-1}(B - A).$$

Proof. Using algebraic manipulations,

$$\begin{aligned}
 2(2A - B)^{-1}B^{-1}(B - A) &= (2A - B)^{-1}B^{-1}(2B - 2A) \\
 &= (2A - B)^{-1}B^{-1}[B - (2A - B)] \\
 &= (2A - B)^{-1}\left[I - B^{-1}(2A - B)\right] \\
 &= (2A - B)^{-1} - (2A - B)^{-1}B^{-1}(2A - B) \\
 &= (2A - B)^{-1} - B^{-1}(2A - B)^{-1}(2A - B) \\
 &= (2A - B)^{-1} - B^{-1},
 \end{aligned}$$

and the lemma is proven. \square

It is known that, for any nonsingular $n \times n$ complex matrix A , there exist $n \times n$ unitary matrices U and V , such that $U^*AV = \Sigma$, Σ being the diagonal matrix of the singular values of A , $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$ and $A = U\Sigma V^*$. Moreover, U^* is the conjugate transpose of U . Then, $X_k = VD_kU^*$, that is $D_k = V^*X_kU$, $D_k^{-1} = U^*X_k^{-1}V$, and $X_k^{-1} = UD_k^{-1}V^*$, and we apply them in Equation (5).

$$VD_{k+1}U^* = VD_kU^* - 2[(2VD_kU^* - VD_{k-1}U^*)^{-1} - UD_{k-1}^{-1}V^*]^{-1}W_1W_2,$$

where $W_1 = UD_k^{-1}V^* - U\Sigma V^*$ and $W_2 = VD_kU^* - VD_{k-1}U^*$. If the initial approximations X_{-1}, X_0 satisfy $D_0 = V^*X_0U$ and $D_{-1} = V^*X_{-1}U$ are diagonal matrices, then all the matrices D_k are diagonal and $D_iD_j = D_jD_i$, for all i, j . Applying several algebraic manipulations on the last equation, we can ensure:

$$\begin{aligned}
 VD_{k+1}U^* &= VD_kU^* - 2[U(2D_k - D_{k-1})^{-1}V^* - UD_{k-1}^{-1}V^*]^{-1}U(D_k^{-1} - \Sigma)V^*V(D_k - D_{k-1})U^* \\
 &= VD_kU^* - 2V[(2D_k - D_{k-1})^{-1} - D_{k-1}^{-1}]^{-1}U^*U(D_k^{-1} - \Sigma)V^*V(D_k - D_{k-1})U^*.
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 D_{k+1} &= D_k - 2[(2D_{k-1} - D_k)^{-1} - D_{k-1}^{-1}]^{-1}(D_k^{-1} - \Sigma)(D_k - D_{k-1}) \\
 &= D_k - 2[(2D_{k-1} - D_k)^{-1} - D_{k-1}^{-1}]^{-1}(D_k - D_{k-1})(D_k^{-1} - \Sigma).
 \end{aligned}$$

Using Lemma 1, we obtain:

$$\begin{aligned}
 [(2D_{k-1} - D_k)^{-1} - D_{k-1}^{-1}]^{-1} &= [2(2D_k - D_{k-1})^{-1}D_{k-1}^{-1}(D_{k-1} - D_k)]^{-1} \\
 &= \frac{1}{2}(2D_k - D_{k-1})D_{k-1}(D_{k-1} - D_k)^{-1}.
 \end{aligned}$$

Then,

$$\begin{aligned}
 D_{k+1} &= D_k - (2D_k - D_{k-1})(D_{k-1} - D_k)^{-1}(D_k^{-1} - \Sigma)(D_k - D_{k-1})D_{k-1} \\
 &= D_k + (2D_k - D_{k-1})D_k^{-1}(I - D_k\Sigma)D_{k-1}.
 \end{aligned}$$

Finally,

$$X_{k+1} = X_k + (2X_k - X_{k-1})X_k^{-1}(I - X_kA)X_{k-1}. \tag{6}$$

In this iterative expression, still, an inverse matrix appears. Therefore, Kurchatov’s scheme is not directly transferable for the calculation of the matrix inverses.

Therefore, we propose a slight change in the Kurchatov divided difference scheme:

$$f[x_k, 2x_{k-1} - x_k]_{KT} = \frac{f(2x_{k-1} - x_k) - f(x_k)}{2(x_{k-1} - x_k)},$$

obtaining a new iterative scheme,

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, 2x_{k-1} - x_k]_{KT}} = x_k - \frac{2(x_{k-1} - x_k)f(x_k)}{f(2x_{k-1} - x_k) - f(x_k)}.$$

Now, we apply this iterative procedure to the matrix equation $F(X) = X^{-1} - A$:

$$\begin{aligned} X_{k+1} &= X_k - 2(F(2X_{k-1} - X_k) - F(X_k))^{-1}F(X_k)(X_{k-1} - X_k) \\ &= X_k - 2\left[(2X_{k-1} - X_k)^{-1} - X_k^{-1}\right]^{-1}\left(X_k^{-1} - A\right)(X_{k-1} - X_k). \end{aligned}$$

Using the transformations defined above and Lemma 1, we obtain:

$$\begin{aligned} VD_{k+1}U^* &= VD_kU^* + 2[(2VD_{k-1}U^* - VD_kU^*)^{-1} - UD_k^{-1}V^*]^{-1}W_1W_2 \\ &= V\left\{D_k - 2[(2D_{k-1} - D_k)^{-1} - D_k^{-1}]^{-1}(D_k^{-1} - \Sigma)(D_{k-1} - D_k)\right\}U^*, \end{aligned}$$

and then,

$$\begin{aligned} D_{k+1} &= D_k - 2[(2D_{k-1} - D_k)^{-1} - D_k^{-1}]^{-1}(D_k^{-1} - \Sigma)(D_{k-1} - D_k) \\ &= 2D_{k-1} - (D_k - 2D_{k-1})\Sigma D_k. \end{aligned}$$

Finally,

$$D_{k+1} = 2D_{k-1} - (D_k - 2D_{k-1})\Sigma D_k \tag{7}$$

From this expression, by using again the transformations defined above, we have

$$\begin{aligned} V^*X_{k+1}U &= 2V^*X_{k-1}U - (V^*X_kU - 2V^*X_{k-1}U)U^*AVV^*X_kU \\ &= V^*[2X_{k-1} - (X_k - 2X_{k-1})AX_k]U, \end{aligned}$$

and

$$X_{k+1} = 2X_{k-1} - (2X_{k-1} - X_k)AX_k, \tag{8}$$

where three matrix products appear, but there are no inverse operators. Equation (8) corresponds to the iterative scheme of the modified Kurchatov-type method for matrix inversion, which we call MKTM.

2.1. Convergence Analysis

Let us consider now two $n \times n$ unitary matrices U and V , satisfying $\Sigma = U^*AV = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, where the singular values $\sigma_i, i = 1, 2, \dots, n$ of A fulfill $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$. We define again $D_k = V^*X_kU, X_k = VD_kU^*$. By using Equation (7), we obtain:

$$D_{k+1} = 2D_{k-1} - (2D_{k-1} - D_k)D_k\Sigma.$$

Next, we demonstrate the order of convergence of the iterative procedure MKTM.

Theorem 3. Let $A \in \mathbb{C}^{n \times n}$ be a nonsingular matrix and X_{-1} and X_0 be the initial approximation such that matrices $V^*X_{-1}U$ and V^*X_0U are diagonal. Then, sequence $\{X_k\}$ obtained by means of the iterative expression (8), where $\Sigma = U^*AV$ is the singular-value decomposition of A , converges to the inverse A^{-1} , with convergence order 1.6180.

Proof. Let us denote $d_k = \text{diag}(d_k^1, d_k^2, \dots, d_k^n)$. By means of component-by-component calculations, we obtain:

$$d_{k+1}^j = 2d_{k-1}^j - (2d_{k-1}^j - d_k^j)d_k^j\sigma_j, \quad j = 1, 2, \dots, n. \tag{9}$$

Let us also denote $e_k^j = d_k^j - 1/\sigma_j$; by subtracting $\frac{1}{\sigma_j}$ from the both sides of the last equation, we obtain:

$$\begin{aligned} e_{k+1}^j &= 2d_{k-1}^j - (2d_{k-1}^j - d_k^j)d_k^j\sigma_j - \frac{1}{\sigma_j} \\ &= \sigma_j \left(\frac{2d_{k-1}^j}{\sigma_j} - 2d_{k-1}^j d_k^j + (d_{k-1}^j)^2 - \frac{1}{\sigma_j^2} \right) \\ &= \sigma_j \left[-2d_{k-1}^j \left(d_k^j - \frac{1}{\sigma_j} \right) + \left(d_k^j - \frac{1}{\sigma_j} \right) \left(d_k^j + \frac{1}{\sigma_j} \right) \right] \\ &= -2\sigma_j e_{k-1}^j e_k^j + (e_k^j)^2 \sigma_j. \end{aligned}$$

This result shows that, for each $j = 1, 2, \dots, n$, d_{k+1} in Equation (9) converges to $\frac{1}{\sigma_j}$ with order of convergence $(1 + \sqrt{5})/2 \approx 1.6180$, which is the only positive root of $\lambda^2 - \lambda - 1 = 0$ (see Theorem 1). In this way, for each $j = 1, 2, \dots, n$, there exists $\{C_k^j\}_k$ satisfying $C_k^j > 0, \forall k$. Moreover, C_k^j tends to zero when $k \rightarrow \infty$. Indeed, $|e_{k+1}^j| \leq C_k^j |(e_k^j)^2|, j = 1, 2, \dots, n$. Furthermore,

$$\|D_{k+1} - \Sigma^{-1}\|_2^2 = \sum_{j=1}^n (e_k^j)^2 \leq \sum_{j=1}^n (C_k^j)^2 (e_k^j)^2 \leq nM_k^2 \|D_k - \Sigma^{-1}\|_2^2,$$

with $M_k = \max_{1 \leq j \leq n} \{C_k^j\}$. Using this result, we obtain:

$$\begin{aligned} \|X_{k+1} - A^{-1}\|_2 &= \|VD_{k+1}U^* - V\Sigma^{-1}U^*\|_2 = \|V(D_{k+1} - \Sigma^{-1})U^*\|_2 \\ &\leq M_k \sqrt{n} \|D_k - \Sigma^{-1}\|_2 \\ &\leq M_k \sqrt{n} \|X_k - A^{-1}\|_2. \end{aligned}$$

Therefore, we can affirm that $\{X_k\}$ converges to A^{-1} . □

To demonstrate the stability of the modified Kurchatov-type iterative scheme, we used the definition introduced by Higham in [14] for the stability of the iterative scheme $Z_{k+1} = H(Z_k)$, with a fixed point Z_* . Assuming that H is Frechét differentiable in fixed point Z_* , then the process is stable in a neighborhood of Z_* if there exists a positive constant C such that $\|H'(Z_*)^k\| \leq C, \forall k > 0$.

Theorem 4. *The modified Kurchatov-type iterative scheme to estimate the inverse of a nonsingular complex matrix defined by expression (8) is stable.*

Proof. The modified Kurchatov-type method can be written as follows:

$$KT(Z_k) = KT \begin{pmatrix} X_k \\ X_{k-1} \end{pmatrix} = \begin{bmatrix} 2X_{k-1} + (2X_{k-1} - X_k)AX_k \\ X_k \end{bmatrix}.$$

So, denoting by $P = (P_1, P_2)^T$, we conclude

$$KT' \begin{pmatrix} X_k \\ X_{k-1} \end{pmatrix} P = \begin{bmatrix} 2X_{k-1}AP_1 - AX_kP_1 - X_kAP_1 + 2P_2 - 2AX_kP_2 \\ P_1 \end{bmatrix}.$$

Now, as $Z = Z^* = (A^{-1}, A^{-1})^T$, we get

$$KT'(Z^*)P = \begin{bmatrix} 0 & 0 \\ I & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 0 \\ P_1 \end{bmatrix}$$

Therefore, matrix $KT'(Z^*)$ is idempotent, and (8) is a stable iterative process. \square

2.2. Qualitative Analysis of Kurchatov-Type Scheme

Now, we considered the Kurchatov-type scheme as a scalar iterative procedure for solving nonlinear equations, analyzed its convergence, and studied its stability. In this analysis, the stability is understood as the dependence on the initial estimations. It is made by applying the vectorial real dynamical concepts to the Kurchatov-type scheme, whose expression is:

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, 2x_{k-1} - x_k]}, \quad k = 0, 1, 2, \dots$$

In the following result, we show that this scheme for solving nonlinear scalar equations $f(x) = 0$ has superlinear convergence.

Theorem 5. *Let us consider a sufficiently differentiable function $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ in an open neighborhood I of the simple root α of nonlinear equation $f(x) = 0$. Furthermore, let us assume that $f'(x)$ is continuous at α and that x_0 and α are near enough between them. Then, sequence x_k , $k = 0, 1, 2, \dots$, generated by the Kurchatov-type scheme, converges to α with order of convergence $\frac{1+\sqrt{5}}{2}$, its error equation being:*

$$e_{k+1} = 2c_2e_{k-1}e_k + \mathcal{O}_2(e_{k-1}, e_k),$$

where $\mathcal{O}_2(e_{k-1}, e_k)$ means that the following terms in the error equation depend on the powers of errors e_{k-1} and e_k such that the sum of the exponents is at least two; moreover, $e_k = x_k - \alpha$ and $c_q = \frac{1}{q!} \frac{f^{(q)}(\alpha)}{f'(\alpha)}$, $q \geq 2$.

Proof. It is known that the Taylor expansion of $f(x_k)$ around α is:

$$f(x_k) = f'(\alpha) [e_k + c_2e_k^2 + c_3e_k^3] + \mathcal{O}(e_k^3).$$

By using the Genocchi–Hermite formula (see [15]) and the expansion of $f'(x + th)$ in the Taylor series around x ,

$$f[x, x + h] = \int_0^1 f'(x + th) dt = f'(x) + \frac{1}{2!} f''(x)h + \frac{1}{3!} f'''(x)h^2 + \mathcal{O}(h^3), \quad (10)$$

with $h = 2x_{k-1} - 2x_k = 2(e_{k-1} - e_k)$, the expansion of the divided difference is:

$$f[x_k, 2x_{k-1} - x_k] = f'(\alpha) [1 + 2c_2e_k + 3c_3e_k^2 + (c_2 + 3c_3e_k)h + 4c_3h] + \mathcal{O}(h^3).$$

Then, the expression of the difference divided as a function of the errors e_{k-1} and e_k is:

$$f[x_k, 2x_{k-1} - x_k] = f'(\alpha) [1 + 2c_2e_{k-1} - 2c_3e_{k-1}e_k + 4c_3e_{k-1}^2 + c_3e_k^2] + \mathcal{O}_3(e_{k-1}, e_k).$$

Therefore,

$$\frac{f(x_k)}{f[x_k, 2x_{k-1} - x_k]} = e_k - 2c_2e_{k-1}e_k + \mathcal{O}_2(e_{k-1}, e_k).$$

By applying Theorem 1, the only positive real root of $\lambda^2 - \lambda - 1 = 0$ (where the coefficients correspond to the exponents of e_{k-1} and e_k in the error equation) is the convergence order of the method, that is $\frac{1+\sqrt{5}}{2}$. \square

From now on, we denote by KT the fixed point operator associated with the Kurchatov-type method applied on $p(z) = z^2 - 1$. As it does not use derivatives, it is not possible to establish a scaling theorem. Therefore, we cannot make the analysis on generic second-degree polynomials.

The fixed point operator depends on two variables: x_k (denoted by x) and x_{k-1} (denoted by z). Therefore,

$$KT(z, x) = \left(x, -\frac{x^2 - 2xz - 1}{2z} \right).$$

Let us analyze now the qualitative performance of the rational operator by means of the asymptotic behavior of the fixed points and the existence of free critical points.

Theorem 6. *The only fixed points of rational operator KT are the roots $x = z = \pm 1$, both being superattracting. Moreover, KT has no free critical points.*

Proof. We solve equation:

$$KT(z, x) = (x, x) \Leftrightarrow x = z \text{ and } x = -\frac{x^2 - 2xz - 1}{2z} \Leftrightarrow x = \pm 1,$$

finding that the only fixed points are $(-1, -1)$ and $(1, 1)$. To study their stability, we considered:

$$KT'(z, x) = \begin{pmatrix} 0 & 1 \\ \frac{(-1+x)(1+x)}{2z^2} & -\frac{x-z}{z} \end{pmatrix}$$

and therefore, its eigenvalues are:

$$\lambda_1(z, x) = \frac{z - x - \sqrt{-2 + 3x^2 - 2xz + z^2}}{2z}$$

and:

$$\lambda_2(z, x) = \frac{z - x + \sqrt{-2 + 3x^2 - 2xz + z^2}}{2z}.$$

As $\lambda_1(-1, -1) = \lambda_2(1, 1) = (0, 0)$, we concluded that fixed points $(1, 1)$ and $(-1, -1)$ are superattracting.

As an immediate consequence of this analysis of $KT'(z, x)$ and its eigenvalues $\lambda_i(z, x)$, $i = 1, 2$, the only $(z, x) \in \mathbb{R}^2$ satisfying $\lambda_1(z, x) = \lambda_2(z, x) = (0, 0)$ are those $(z, x) = (\pm 1, \pm 1)$, where $z = x$. Then, they are the only critical points, and there do not exist free critical points. \square

From this results, we concluded that no other performance than convergence to the roots is possible, when the Kurchatov-type is applied on $p(x)$, as any other basin of attraction would need a free critical point inside.

In Figure 1, we show the dynamical plane of the KT operator, x and z being real, corresponding to the abscissa and ordinate axis, respectively (see [22] for the routines). The mesh used has 1000×1000 points, and the maximum amount of iterations is 40. The distance to the root lower than 10^{-3} is the stopping criterion.

Then, if each point of the mesh is considered as a seed of the method, when it converges to one of the roots of $p(z)$ (located in $x = z$), it is represented in orange or green color, depending on the root it has converged to. The color is brighter the lower the amount of iterations is. When the initial estimation, after the iterative process, reaches 40 iterations with no convergence, it is colored in black.

In Figure 1, we observe that the basins of attraction of both roots of $p(z)$ have symmetric shapes. In spite of the only basins of attraction being those of the roots (Theorem 6), there are black areas in the dynamical planes. They correspond to areas of slow convergence, whose initial estimations converge with a higher number of iterations.

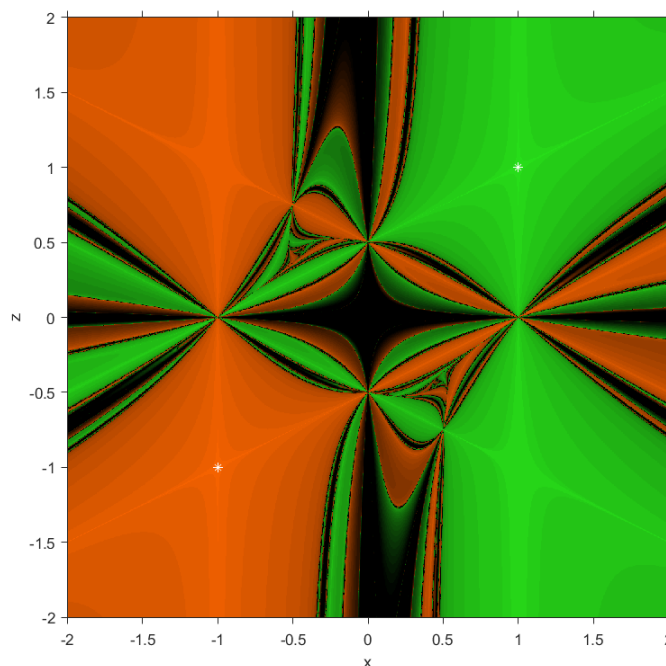


Figure 1. Dynamical plane of the Kurchatov-type scheme on $p(z) = z^2 - 1$.

In [19,20], the stability analysis of the secant and Steffensen with memory schemes was performed, among other iterative schemes with memory. In both cases, the dynamical planes plotted showed full convergence to the roots, without black areas of slow convergence or divergence. Later on, we checked how the differences among the Kurchatov-type and secant or Steffensen with memory stability affected their numerical performance.

3. Steffensen with Memory

Now, we considered the Traub–Steffensen family [18], whose iterative expression is:

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k + \gamma f(x_k), x_k]}, \tag{11}$$

where γ is an arbitrary constant different from zero. Furthermore, if $\gamma = 1$ in this iterative scheme, we obtain the well-known Steffensen method [17]. It is easy to show that this iterative procedure has order of convergence two for any value of γ , its error equation being:

$$e_{k+1} = c_2(1 + \gamma f'(\alpha))e_k^2 + \mathcal{O}[e_k^3], \tag{12}$$

where $e_k = x_k - \alpha$ represents the error in each iteration and $c_j = \frac{1}{j!} f^{(j)}(\alpha) / f'(\alpha)$, $j \geq 2$. Let us also remark that this is a derivative-free scheme without memory, because it uses only the information of the current iteration. Studying the equation of the error (12) of family (11), we notice that, if $\gamma = -1/f(\alpha)$, the iterative scheme (11) achieves the third order of convergence. This value of the parameter cannot be used, as α is unknown. Therefore, the approximation of $f'(\alpha)$ as the first-order divided differences $\gamma = -1/f[x_k, x_{k-1}]$ is employed, obtaining $x_k + \gamma f(x_k) = x_k - f(x_k)/f[x_k, x_{k-1}]$, which is the secant scheme, and (11) is an iterative procedure with memory.

3.1. Iterative Scheme Design for the Matrix Inverse

Our aim now was to adapt Steffensen’s scheme with memory to solve the matrix Equation (1). In [23] Monsalve et al. adapted the secant method to estimate the solution

of Equation (1) for A diagonalizable, and in [13], the authors extended this result to any nonsingular matrix by means of the expression:

$$W_{k+1} = W_{k-1} + W_k - W_{k-1}AW_k, \quad k \geq 0, \tag{13}$$

where W_0 and W_{-1} are the seeds.

In a similar way, Expression (11) for matrix equations takes the form:

$$\begin{aligned} X_{k+1} &= X_k - [F(X_k) - F(W_k)]^{-1}F(X_k)(X_k - W_k) \\ &= X_k - [X_k^{-1} - W_k^{-1}]^{-1}(X_k^{-1} - A)(X_k - X_{k-1}), \end{aligned} \tag{14}$$

which includes the inverse calculations for the estimation of A^{-1} . This circumstance must be avoided.

As A is a nonsingular $n \times n$ complex matrix, there exist $n \times n$ unitary matrices U and V , such that $\Sigma = U^*AV = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$ are the singular values of A . We define $X_k = VD_kU^*$ and $X_k^{-1} = UD_k^{-1}V^*$. Then, Equation (14) takes the form:

$$VD_{k+1}U^* = VD_kU^* - [UD_k^{-1}V^* - UD_k^{-1}V^*]^{-1}(UD_k^{-1}V^* - U\Sigma V^*)(VD_kU^* - VD_{k-1}U^*) \tag{15}$$

If the initial approximations X_{-1} and X_0 satisfy $D^{-1} = V^*X^{-1}U$ and $D_0 = V^*X_0U$, then D_k are diagonal matrices and $D_iD_j = D_jD_i$ for all i, j . By applying several algebraic manipulations on Equation (15), we can ensure:

$$\begin{aligned} VD_{k+1}U^* &= VD_kU^* - [UD_k^{-1}V^* - (VD_{k-1}U^* + VD_kU^* - VD_{k-1}U^*U\Sigma V^*VD_kU^*)^{-1}]^{-1} \\ &\quad (UD_k^{-1}V^* - U\Sigma V^*)(VD_{k-1}U^* + U\Sigma V^*VD_kU^* - VD_{k-1}U^*) \\ &= V\{D_k - [D_k^{-1} - (D_{k-1} + D_k - D_{k-1}\Sigma D_k)^{-1}]^{-1}(D_k^{-1} - \Sigma)D_{k-1}(\Sigma D_k - I)\}U^*, \end{aligned} \tag{16}$$

and therefore,

$$\begin{aligned} D_{k+1} &= D_k - [D_k^{-1} - (D_{k-1} + D_k - D_{k-1}\Sigma D_k)^{-1}]^{-1}(D_k^{-1} - \Sigma)D_{k-1}(\Sigma D_k - I) \\ &= D_{k-1} + (I - D_{k-1}\Sigma)(2I - D_k\Sigma)D_k, \end{aligned} \tag{17}$$

where I denotes the identity matrix of size n . Now, taking $D_k = V^*X_kU$ and $D_k^{-1} = U^*X_k^{-1}V$ and after more algebraic manipulations, we obtain:

$$X_{k+1} = X_{k-1} + (I - X_{k-1}A)(2I - X_kA)X_k, \quad k = 1, 2, \dots, \tag{18}$$

given the initial approximations X_0 and X_{-1} . We denote this iterative scheme with memory as *SMM*. Let us remark that Expression (18) does not include inverse operators, as intended.

3.2. Convergence Analysis and Stability

We present in the next result the convergence analysis of the iterative method given by Expression (18).

Theorem 7. Let $U^*AV = \Sigma$ be the singular-value decomposition of the a nonsingular matrix $A \in \mathbb{C}^{n \times n}$. Furthermore, let X_0 and X_{-1} be the initial approximations to A^{-1} , such that $V^*X_{-1}U$ and V^*X_0U are diagonal matrices. Then, sequence $\{X_k\}$, generated by (18), converges to A^{-1} with order of convergence $1 + \sqrt{2} \approx 2.4142$.

Proof. Let U and V be unitary matrices such that $U^*AV = \Sigma$, Σ being the diagonal matrix of the singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$ of A . Furthermore, by denoting $D_k = V^*X_kU$ and $X_k = VD_kU^*$ for $k = -1, 0, 1, \dots$, from Equation (18), we obtain:

$$VD_{k+1}U^* = VD_{k-1}U^* + (VIV^* - VD_{k-1}U^*U\Sigma V^*)(2VIV^* - VD_kU^*U\Sigma V^*)VD_kU^* \\ = V[D_{k-1} + (I - D_{k-1}\Sigma)(2I - D_k\Sigma)D_k]U^*.$$

Let us remark that $D_k = \text{diag}(d_k^1, d_k^2, \dots, d_k^n)$, for $k = -1, 0, 1, \dots$. Therefore,

$$D_{k+1} = D_{k-1} + (I - D_{k-1}\Sigma)(2I - D_k\Sigma)D_k. \tag{19}$$

Therefore, in Equation (19), using component-by-component calculations, we obtain:

$$d_{k+1}^j = d_{k-1}^j + (1 - d_{k-1}^j\sigma_j)(2 - d_k^j\sigma_j)d_k^j \dots j = 1, 2, \dots, n. \tag{20}$$

We denote $e_k^j = d_k^j - 1/\sigma_j$. We subtract $\frac{1}{\sigma_j}$ from both sides of Equation (20), and we obtain:

$$e_{k+1}^j = e_{k-1}^j + \sigma_j \left(\frac{1}{\sigma_j} - d_{k-1}^j \right) \sigma_j \left(\frac{2}{\sigma_j} - d_k^j \right) d_k^j \\ = \sigma_j^2 (e_k^j)^2 e_{k-1}^j. \tag{21}$$

Theorem 1 allows us to conclude that, for each j from 1 to n , d_{k+1} converges to $\frac{1}{\sigma_j}$ with the convergence order the only positive root of $\lambda^2 - 2\lambda - 1 = 0$, $\lambda = 1 + \sqrt{2} \approx 2.4142$ (see [15]). In this way, for each $j = 1, 2, \dots, n$, there exist a sequence $\{C_k^j\}_k$ satisfying $C_k^j > 0$, $\forall k$, and $\{C_k^j\}_k \rightarrow 0$ when $k \rightarrow \infty$. Indeed, $|e_{k+1}^j| \leq C_k^j |(e_k^j)|^2$, $1 \leq j \leq n$. On the other hand,

$$\|D_{k+1} - \Sigma^{-1}\|_2^2 = \sum_{j=1}^n (e_k^j)^2 \leq \sum_{j=1}^n (C_k^j)^2 [(e_k^j)^2]^2 \leq nM_k^2 \|D_k - \Sigma^{-1}\|_2^2,$$

where $M_k = \max_{1 \leq j \leq n} \{C_k^j\}$. Using this result, we obtain:

$$\|X_{k+1} - A^{-1}\|_2 = \|V(D_{k+1} - \Sigma^{-1})U^*\|_2 \\ \leq \|V\|_2 \|D_{k+1} - \Sigma^{-1}\|_2 \|U^*\|_2 = \|D_{k+1} - \Sigma^{-1}\|_2 \\ \leq M_k \sqrt{n} \|D_k - \Sigma^{-1}\|_2 \\ \leq M_k \sqrt{n} \|X_k - A^{-1}\|_2.$$

Then, $\{X_k\}$ converges to A^{-1} , and the proof is finished. \square

In the following result, we prove the stability of this scheme.

Theorem 8. *Steffensen’s method with memory to estimate the inverse of a given matrix with the expression (18) is a stable iterative scheme.*

Proof. Steffensen’s scheme with memory can be written as a fixed point scheme as follows:

$$S(Z_k) = S \left(\begin{matrix} X_k \\ X_{k-1} \end{matrix} \right) = \left[\begin{matrix} X_{k-1} + (I - X_{k-1}A)(2I - X_kA)X_k \\ X_k \end{matrix} \right].$$

Then, given $P = (P_1, P_2)^T$, we deduce that:

$$S' \begin{pmatrix} X_k \\ X_{k-1} \end{pmatrix} P = \begin{bmatrix} P_1 - A(2I - X_k A)X_k P_1 - (I - X_{k-1} A)AX_k P_2 + (I - X_{k-1} A)(2I - X_k A)P_2 \\ P_1 \end{bmatrix}.$$

Next, for $Z = Z^* = (A^{-1}, A^{-1})^T$, we have:

$$S'(Z^*)P = \begin{bmatrix} 0 & 0 \\ I & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 0 \\ P_1 \end{bmatrix};$$

therefore, $S'(Z^*)$ is an idempotent matrix, and the iterative process (18) is stable. \square

4. Approximating the Moore–Penrose Inverse

For a complex matrix that is not necessarily square, the pseudo-inverse can be defined as an object similar to an inverse matrix. Given a complex matrix, it is possible to define many possible pseudo-inverses, but the most-frequent one is the Moore–Penrose inverse.

In cases when a system of equations has no solution, there is no inverse of the matrix of the coefficients that defines the system. In these cases, it may be useful to find a value that is an approximate solution in terms of error minimization, such as being able to find the best fit of a dataset using the pseudo-inverse (Moore–Penrose inverse) of the coefficient matrix. See, for example, the classical texts of Ben-Israel [24] and Higham [14].

We now extend the proposed iterative methods of this manuscript for calculating the Moore–Penrose inverse A^\dagger of an $m \times n$ complex matrix A . This is the unique $n \times m$ matrix X that satisfies the following conditions:

$$AXA = A, \quad XAX = X, \quad (AX)^* = AX, \quad (XA)^* = XA.$$

A computationally simple and accurate way to compute the Moore–Penrose inverse is through the use of the singular-value decomposition of A . Then, if the rank of A is $r \leq \min(m, n)$, we obtain:

$$A = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^* \tag{22}$$

where $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, with $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ a unitary matrix. Furthermore, it is known that:

$$A^\dagger = V^* \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} U, \tag{23}$$

where $\Sigma^{-1} = \text{diag}(1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_r)$.

Theorem 9. Let $A \in \mathbb{C}^{m \times n}$ be a matrix with singular-value decomposition satisfying (22) whose $\text{rank}(A) = r$. If X_{-1} and X_0 are the initial estimations such that:

$$V^* X_{-1} U = \begin{bmatrix} \Sigma_{-1} & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad V^* X_0 U = \begin{bmatrix} \Sigma_0 & 0 \\ 0 & 0 \end{bmatrix},$$

given that Σ_{-1} and Σ_0 are diagonal matrices of size $r \times r$, then the sequences $\{X_k\}$, generated by Equation (18) using SSM and (8) using MKTM, converge to A^\dagger with order of convergence $1 + \sqrt{2}$ and $(1 + \sqrt{5})/2$, respectively.

Proof. Given the singular-value decomposition of A (see (22)), we define the matrix D_k :

$$D_k = V^* X_k U = \begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix},$$

for any fixed arbitrary value of k , with $\Sigma_k \in \mathbb{C}^{r \times r}$. Now, by using the iterative expression in Equation (18) and (8), we obtain:

$$\begin{bmatrix} \Sigma_{k+1} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \Sigma_{k-1} + (I - \Sigma_{k-1}\Sigma)(2I - \Sigma_k\Sigma)\Sigma_k & 0 \\ 0 & 0 \end{bmatrix},$$

and

$$\begin{bmatrix} \Sigma_{k+1} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2\Sigma_{k-1} - (2\Sigma_{k-1} - \Sigma_k)\Sigma_k\Sigma & 0 \\ 0 & 0 \end{bmatrix},$$

respectively. Because Σ_{-1} and Σ_0 are diagonal matrices, then all matrices Σ_k are diagonal as well. Then, the expression $\Sigma_{k+1} = \Sigma_{k-1} + (I - \Sigma_{k-1}\Sigma)(2I - \Sigma_k\Sigma)\Sigma_k$ and $\Sigma_{k+1} = 2\Sigma_{k-1} - (2\Sigma_{k-1} - \Sigma_k)\Sigma_k\Sigma$ represents r scalar uncoupled iterations converging to $\frac{1}{\sigma_i}$, $1 \leq i \leq r$, with order of convergence $1 + \sqrt{2}$ using SSM and $(1 + \sqrt{5})/2$ using MKTM, i.e.,

$$\|\Sigma_{k+1} - \Sigma^{-1}\|_2^2 \leq rM_k^2 \|\Sigma_k - \Sigma^{-1}\|_2^2,$$

with $M_k = \max_{1 \leq i < r} \{C_k^2\}$, $C_k^{-1} > 0$, such that $\{e_k^i\}$ tends to zero if k tends to infinity. Just as was argued in (8), we have that:

$$\|X_{k+1} - A^\dagger\|_2 \leq \sqrt{r}M_k \|X_k - A^\dagger\|_2.$$

Therefore, we can say that $\{X_k\}$ converges to A^\dagger , with the desired order of convergence. \square

5. Numerical Experiments

In this section, we present the numerical tests of the behavior of the Steffensen method with memory (SMM) and Kurchatov type's (MKTM) designed to calculate the inverse and the Moore–Penrose inverse applied to different matrices. For comparison, we used the Newton–Schulz method (NS) [14], and the secant method (SM) [13]. The numerical calculations were made with MATLAB 2022a (MathWorks, USA) using a 3 GHz 10-Core Intel Xeon W processor, 64 GB 2666 MHz DDR4 (iMac Pro). As stopping criteria for all numerical tests, we used $\|X_{k+1} - X_k\|_2 < 10^{-6}$ or $\|F(X_{k+1})\|_2 < 10^{-6}$, $F(X) = 0$ being the nonlinear matrix equation to be solved for estimating the inverse or pseudo-inverse of a complex matrix A .

In addition, when verifying the theoretical results numerically, we used the order of approximate computational convergence (COC) introduced by Jay [25], defined as:

$$\text{order} \approx \text{COC} = \frac{\ln(\|F(X_{k+1})\|_2 / \|F(X_k)\|_2)}{\ln(\|F(X_k)\|_2 / \|F(X_{k-1})\|_2)}.$$

Another form of the numerical approximation of the order of theoretical convergence presented by the authors in [26], denoted as ACOC, is defined as:

$$\text{order} \approx \text{ACOC} = \frac{\ln(\|X_{k+1} - X_k\|_2 / \|X_k - X_{k-1}\|_2)}{\ln(\|X_k - X_{k-1}\|_2 / \|X_{k-1} - X_{k-2}\|_2)}.$$

To show the order of convergence of the methods in the numerical tests, we used any of these estimates of computational order. In the tables, we write “-” when the COC (or ACOC) vector is unstable. Furthermore, the mean of elapsed time after 50 executions of the codes appears in the tables, calculated by using the CPU-time command.

Example 1. As a first example, we looked for the inverse random matrices of size $n \times n$, where $n = 10, 100, 200, 300, 400$, and 500. Since Newton–Schulz’s method needs one initial point, we took it as $X_0 = A^T / \|A\|^2$. The rest of the methods need two initial points, which we took as $X_{-1} = A^T / \|A\|^2$ and $X_0 = 0.5X_{-1}$.

Table 1 shows the results obtained by approximating the inverses of nonsingular random matrices of sizes $n = 10, 100, 200, 300, 400,$ and 500 using the Newton–Schulz, secant, Steffensen with memory, and Kurchatov-type methods. The number of iterations, residuals, and the COC value are shown. The results confirmed the theoretical order of convergence obtained from each method. Using all methods, the approximation of the inverse of A was obtained. In all cases, the Steffensen method with memory showed better results in terms of the number of iterations and the computational time. The graphs shown in Figure 2 represent the results presented in Table 1 for the cases of matrices of $n = 10, 300,$ and 500 .

Table 1. Results obtained by approximating the inverse of random matrices of different sizes.

Method	n	Iter	$\ X_{k+1} - X_k\ _2$	$\ F(X_{k+1})\ _2$	COC	E-Time
NS	10	25	7.5×10^{-3}	1.9×10^{-9}	2.0000	0.0151
SM	10	35	7.4×10^{-4}	2.0×10^{-9}	1.6180	0.0141
SMM	10	15	1.2×10^{-1}	2.4×10^{-8}	2.4142	0.0136
MKTM	10	45	1.8×10^{-3}	1.3×10^{-8}	1.6119	0.0147
NS	100	29	2.5×10^{-4}	1.4×10^{-11}	2.0000	0.0660
SM	100	40	1.6×10^{-3}	3.8×10^{-8}	1.6181	0.1043
SMM	100	22	2.4×10^{-2}	5.0×10^{-9}	2.4142	0.0501
KT	100	53	5.7×10^{-4}	9.2×10^{-9}	1.6687	0.1293
NS	200	28	1.6×10^{-4}	4.4×10^{-11}	2.0000	0.2656
SM	200	39	8.3×10^{-5}	1.5×10^{-9}	1.6180	0.3803
SMM	200	24	4.5×10^{-2}	2.6×10^{-7}	2.4142	0.1676
MKTM	200	50	2.6×10^{-3}	5.8×10^{-7}	1.6172	0.5134
NS	300	32	2.9×10^{-2}	1.4×10^{-7}	2.0000	0.6303
SM	300	45	7.0×10^{-3}	2.9×10^{-7}	1.6180	0.8967
SMM	300	25	1.2×10^{-2}	5.9×10^{-10}	2.4142	0.4222
MKTM	300	60	1.6×10^{-4}	9.2×10^{-10}	1.6193	1.3173
NS	400	33	8.8×10^{-4}	1.6×10^{-10}	2.0000	0.9306
SM	400	46	1.5×10^{-3}	2.7×10^{-8}	1.6181	1.6293
SMM	400	25	1.3×10^{-4}	2.5×10^{-7}	2.4142	0.6986
MKTM	400	61	2.6×10^{-3}	9.5×10^{-8}	1.6841	2.4109
NS	500	36	2.5×10^{-2}	2.1×10^{-8}	2.0000	1.8538
SM	500	51	1.4×10^{-3}	5.8×10^{-9}	1.6180	2.8333
SMM	500	28	3.0×10^{-2}	8.0×10^{-10}	2.4076	1.1715
MKTM	500	67	1.1×10^{-2}	3.4×10^{-7}	1.6336	4.1268

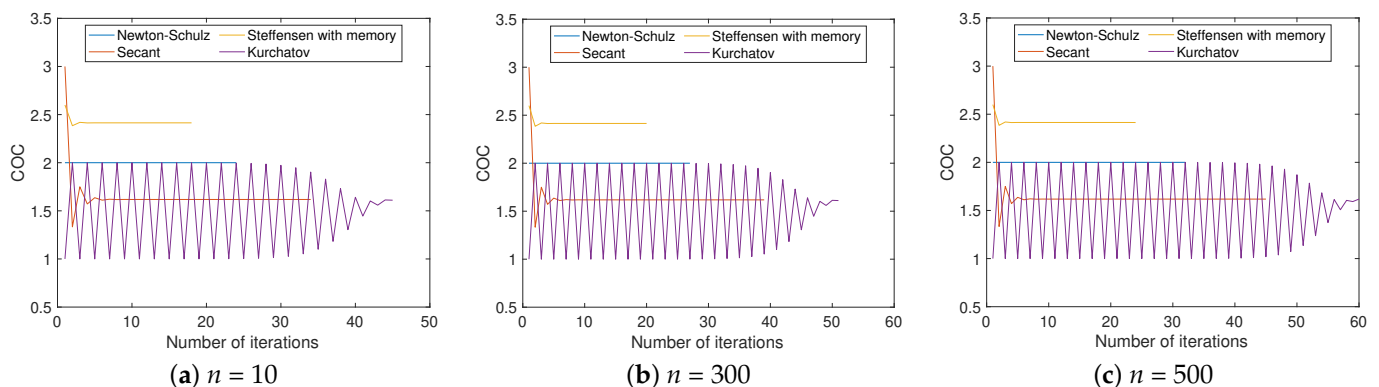


Figure 2. Graphs of the number of iterations vs. COC for three of the cases in Table 1.

Example 2. Now, we built square matrices of size $n \times n$ using different MATLAB functions, such as:
 (a) $A = \text{gallery}('lehmmer', n)$. Symmetric and positive definite matrix of size $n \times n, a_{i,j} = i/j, \forall i, j$.
 (b) $A = \text{gallery}('riemann', n)$. Riemann matrix of size $n \times n$.

- (c) $A = \text{gallery}('ris', n)$. Hankel matrix of size $n \times n$.
- (d) $A = \text{gallery}('grcar', n)$. Toeplitz matrix of size $n \times n$.
- (e) $A = \text{gallery}('leslie', n)$. Leslie matrix of size $n \times n$ with application in problems of population models.
- (f) $A = \text{gallery}('parter', n)$. Parter matrix of size $n \times n$.

Here, we used as stopping criteria $\|X_{k+1} - X_k\|_2 < 10^{-10}$ or $\|F(X_{k+1})\|_2 < 10^{-10}$ and the same initial approximations as in Example 1. The numerical results obtained are shown in Table 2 and Figure 3. As in the previous example, the proposed methods showed a good performance in terms of stability, precision, and number of iterations required.

Table 2. Results obtained by approximating the inverse of classical square matrices of different sizes.

Method	Matrix	n	Iter	$\ X_{k+1} - X_k\ _2$	$\ F(X_{k+1})\ _2$	COC	E-Time
NS	Lehmer	10	19	3.5×10^{-7}	5.7×10^{-15}	-	0.0162
SM	Lehmer	10	26	1.9×10^{-7}	1.7×10^{-13}	1.6177	0.0157
SMM	Lehmer	10	14	7.6×10^{-4}	4.2×10^{-11}	2.4142	0.0149
MKTM	Lehmer	10	33	1.6×10^{-7}	1.8×10^{-13}	1.6384	0.0168
NS	Riemann	100	24	5.0×10^{-8}	9.2×10^{-14}	1.5057	0.0638
SM	Riemann	100	33	1.5×10^{-7}	2.1×10^{-12}	1.5057	0.0800
SMM	Riemann	100	18	7.5×10^{-5}	4.6×10^{-12}	2.3949	0.0482
MKTM	Riemann	100	43	1.8×10^{-8}	1.8×10^{-12}	2.3949	0.1013
NS	Henkel	200	10	3.7×10^{-9}	1.9×10^{-15}	-	0.0850
SM	Henkel	200	13	2.0×10^{-9}	2.5×10^{-15}	1.6070	0.1093
SMM	Henkel	200	7	1.0×10^{-6}	9.5×10^{-16}	2.3550	0.0360
MKTM	Henkel	200	14	3.4×10^{-8}	4.1×10^{-13}	1.5931	0.0418
NS	Toeplitz	300	10	2.5×10^{-9}	1.1×10^{-15}	-	0.1672
SM	Toeplitz	300	13	1.5×10^{-9}	4.3×10^{-15}	1.6165	0.2077
SMM	Toeplitz	300	7	6.2×10^{-6}	2.0×10^{-13}	2.4143	0.1272
MKTM	Toeplitz	300	15	2.5×10^{-10}	9.7×10^{-16}	1.5111	0.2319
NS	Leslie	400	23	4.3×10^{-5}	4.7×10^{-12}	2.0000	0.5788
SM	Leslie	400	32	5.3×10^{-6}	2.3×10^{-11}	1.6180	0.8384
SMM	Leslie	400	18	3.4×10^{-6}	3.1×10^{-16}	2.1993	0.4909
MKTM	Leslie	400	42	5.8×10^{-8}	1.1×10^{-13}	-	1.1811
NS	Parter	500	10	7.5×10^{-8}	3.7×10^{-15}	1.9969	1.3828
SM	Parter	500	13	4.8×10^{-8}	9.8×10^{-13}	1.6181	0.4933
SMM	Parter	500	7	5.1×10^{-5}	2.5×10^{-11}	2.4144	0.2832
MKTM	Parter	500	15	3.4×10^{-8}	8.3×10^{-13}	1.6433	0.5984

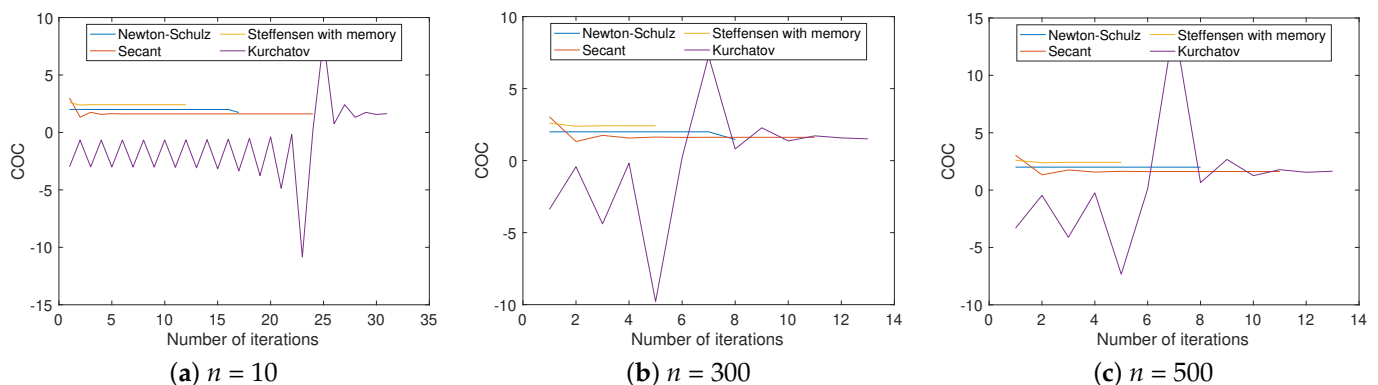


Figure 3. Graphs of the number of iterations vs. COC for three of the sizes in Table 2.

Example 3. Finally, we tested the methods for computing the Moore–Penrose inverse of $m \times n$ random matrices for different values of m and n . The matrices of the initial approximations were

calculated in the same way as in the previous examples, and the stopping criterion was the one used in Example 1.

The results obtained for the number of iterations, the residuals, and the ACOC value of Example 3 are shown in Table 3 and Figure 4. The methods gave us an approximation of the inverse Moore–Penrose and showed the same behavior as in the previous examples.

Table 3. Results obtained to approximate the Moore–Penrose inverse of a rectangular random matrix.

Method	m	n	Iter	$\ X_{k+1} - X_k\ _2$	ACOC	E-Time
NS	20	10	11	1.2×10^{-7}	2.0086	0.0162
SM	20	10	17	9.5×10^{-9}	1.6220	0.0152
SMM	20	10	10	1.6×10^{-10}	2.4159	0.0158
MKTM	20	10	20	3.2×10^{-7}	1.7104	0.0221
NS	200	100	16	1.1×10^{-8}	2.0049	0.0916
SM	200	100	24	4.2×10^{-9}	1.6217	0.1406
SMM	200	100	14	1.3×10^{-12}	2.4147	0.0766
MKTM	200	100	30	1.7×10^{-8}	1.7264	0.1773
NS	300	400	19	3.1×10^{-4}	2.1197	0.3610
SM	300	400	29	8.6×10^{-6}	1.6446	0.5411
SMM	300	400	16	1.7×10^{-4}	2.5496	0.3397
MKTM	300	400	37	7.6×10^{-5}	1.4546	0.7319
NS	500	600	21	3.8×10^{-4}	2.1114	1.0525
SM	500	600	32	3.9×10^{-6}	1.6380	1.7327
SMM	500	600	18	2.7×10^{-6}	2.4445	0.8430
MKTM	500	600	41	7.0×10^{-5}	1.4333	2.3384
NS	1000	900	25	7.2×10^{-13}	2.0001	4.6221
SM	1000	900	36	1.1×10^{-8}	1.6217	6.8167
SMM	1000	900	20	1.2×10^{-7}	2.4249	3.2718
MKTM	1000	900	47	3.6×10^{-7}	-	9.5099

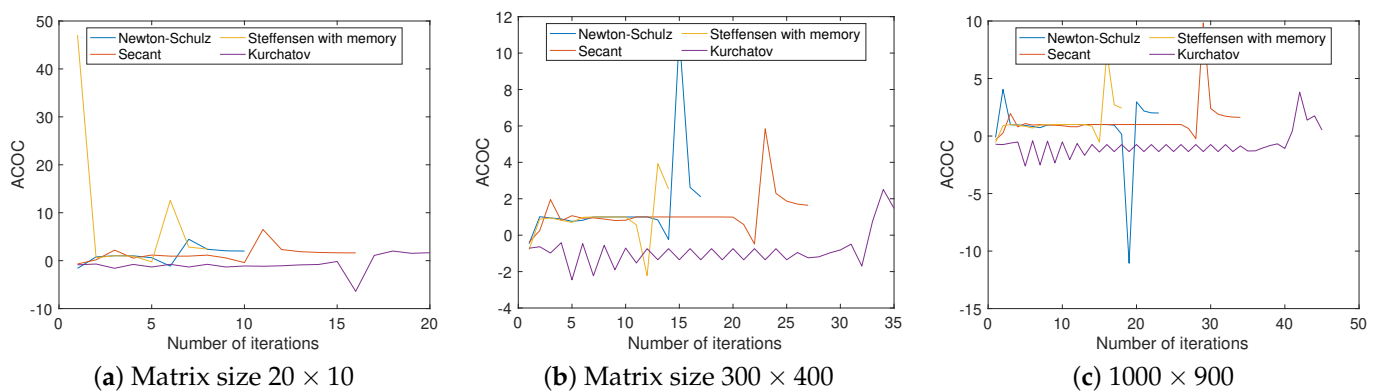


Figure 4. Graphs of the number of iterations vs. ACOC for three of the sizes in Table 3.

6. Conclusions

In this manuscript, we widened the set of iterative methods able to be applied for estimating generalized inverses of complex matrices, by using schemes with memory able to improve the Newton–Schulz scheme. Two procedures with memory were designed for approximating the inverses of nonsingular complex matrices or pseudo-inverses, in the case that the matrices are singular. The order of convergence and stability were proven, and in the case of the Steffensen with memory scheme, its order of convergence improved that of the Newton–Schulz method.

The method using Kurchatov’s divided differences cannot be directly adapted to estimate generalized inverses of complex matrices. To overcome this difficulty, Kurchatov-

type divided differences were used. In this process, there was a decrease in the order of convergence of the starting method and a change in its behavior.

The technique used to adapt the iterative methods can be applied to the resolution of other types of matrix equations with an especially significant role in various areas such as control theory, dynamic programming, ladder networks, statistics, etc.

This research opens new ways in the design of iterative procedures for solving this kind of nonlinear matrix equation, with promising numerical performance, which showed agreement with theoretical results and stable results.

Author Contributions: Conceptualization, J.R.T.; methodology, J.G.M.; software, M.P.V.; formal analysis, J.R.T.; investigation, A.C.; writing—original draft preparation, J.G.M. and M.P.V.; writing—review and editing, A.C. and J.R.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers for their suggestions and comments that have improved the final version of this manuscript.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Torokhti, A.; Soto-Quiros, P. Generalized Brillinger-like transforms. *IEEE Signal Process. Lett.* **2016**, *23*, 843–847. [[CrossRef](#)]
2. Chung, J.; Chung, M. Computing optimal low-rank matrix approximations for image processing. In Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers, IEEE, Pacific Grove, CA, USA, 3–6 November 2013; pp. 670–674.
3. Chountasis, S.; Katsikis, V.N.; Pappas, D. Applications of the Moore–Penrose inverse in digital image restoration. *Math. Probl. Eng.* **2009**, *2009*, 170724. [[CrossRef](#)]
4. Miljković, S.; Miladinović, M.; Stanimirović, P.; Stojanović, I. Application of the pseudoinverse computation in reconstruction of blurred images. *Filomat* **2012**, *26*, 453–465. [[CrossRef](#)]
5. Liu, J.; Zhang, H.; Jia, J. Cryptanalysis of schemes based on pseudoinverse matrix. *Wuhan Univ. J. Nat. Sci.* **2016**, *21*, 209–213. [[CrossRef](#)]
6. Dang, V.H.; Nguyen, T.D. Construction of pseudoinverse matrix over finite field and its applications. *Wirel. Pers. Commun.* **2017**, *94*, 455–466. [[CrossRef](#)]
7. Nguyen, C.-T.; Tsai, Y.-W. Finite-time output feedback controller based on observer for the time-varying delayed systems: A Moore–Penrose inverse approach. *Math. Probl. Eng.* **2017**, *2017*, 2808094. [[CrossRef](#)]
8. Ansari, U.; Bajodah, A.H. Robust launch vehicle’s generalized dynamic inversion attitude control. *Aircr. Eng. Aerosp. Technol.* **2017**, *89*, 902–910. [[CrossRef](#)]
9. Weiguo, L.; Juan, L.; Tiantian, Q. A family of iterative methods for computing Moore–Penrose inverse of a matrix. *Linear Algebra Appl.* **2013**, *438*, 47–56. [[CrossRef](#)]
10. Toutounian, F.; Soleymani, F. An iterative method for computing the approximate inverse of a square matrix and the Moore–Penrose inverse of a non-square matrix. *Appl. Math. Comput.* **2013**, *224*, 671–680. [[CrossRef](#)]
11. Stanimirović, P.; Kumar, A.; Katsikis, V. Further efficient hyperpower iterative methods for the computation of generalized inverses $a_{T,S}^{(2)}$. *RACSAM Rev. Real Acad. Cienc. Exactas Fis. Nat. Ser. A Mat.* **2019**, *113*, 3323–3339.
12. Kaur, M.; Kansal, M.; Kumar, S. An efficient hyperpower iterative method for computing weighted Moore–Penrose inverse. *AIMS Math.* **2020**, *5*, 1680–1692. [[CrossRef](#)]
13. Artidiello, S.; Cordero, A.; Torregrosa, J.R.; Vassileva, M.P. Generalized Inverses Estimations by Means of Iterative Methods with Memory. *Mathematics* **2020**, *8*, 2. [[CrossRef](#)]
14. Higham, N.J. *Functions of Matrices: Theory and Computation*; SIAM: Philadelphia, PA, USA, 2008.
15. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solutions of Nonlinear Equations in Several Variables*; Academic Press Inc.: Cambridge, MA, USA, 1970.
16. Kurchatov, V.A. On a method of linear interpolation for the solution of functional equations. *Dokl. Acad. Nauk. SSSR* **1971**, *198*, 524–526. Translation in *Sov. Math. Dokl.* **1971**, *12*, 835–838. (In Russian)
17. Steffensen, J.F. Remarks on iteration. *Scand. Actuar. J.* **1933**, *1*, 64–72. [[CrossRef](#)]
18. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: Hoboken, NJ, USA, 1964.
19. Campos, B.; Cordero, A.; Torregrosa, J.R.; Vindel, P. A multidimensional dynamical approach to iterative methods with memory. *Appl. Math. Comput.* **2015**, *271*, 701–715. [[CrossRef](#)]

20. Campos, B.; Cordero, A.; Torregrosa, J.R.; Vindel, P. Stability analysis of iterative methods with memory. In Proceedings of the 15th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE 2015), Rota Cadiz, Spain, 6–10 July 2015; pp. 285–290.
21. Robinson, R.C. An Introduction to Dynamical Systems, Continuous and Discrete. In *Pure and Applied Undergraduate Texts*; American Mathematical Society: Providence, RI, USA, 2012.
22. Chicharro, F.I.; Cordero, A.; Torregrosa, J.R. Drawing dynamical and parameters planes of iterative families and methods. *Sci. World J.* **2013**, *2013*, 780153. [[CrossRef](#)] [[PubMed](#)]
23. Monsalve, M.; Raydan, M. A secant method for nonlinear matrix problem. *Numer. Linear Algebra Signals Syst. Control.* **2011**, *80*, 387–402. *Lecture Notes in Electrical Engineering*.
24. Ben-Israel, A.; Greville, T.N.E. *Generalized Inverses*; Springer: New York, NY, USA, 2003.
25. Jay, L.O. A note of Q-order of convergence. *BIT Numer. Math.* **2001**, *41*, 422–429. [[CrossRef](#)]
26. Cordero, A.; Hueso, J.L.; Torregrosa, J.R. Variants of Newton's method using fifth-order quadrature formulas. *Appl. Math. Comput.* **2012**, *190*, 686–698. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.