Communications in Advanced Mathematical Sciences
Vol. III, No. 2, 109-114, 2020
Research Article
e-ISSN: 2651-4001
DOI: 10.33434/cams.718365

A Family of Arbitrary High-Order Iterative Methods for Approximating Inverse and the Moore–Penrose Inverse

Esmaeil Kokabifar[1]*

Abstract
In this work, a family of iterative algorithms for approximating the inverse of a square matrix and the Moore-Penrose inverse of a non-square one is proposed. These methods are based on arbitrary high-order iterative techniques which are used for computing roots of a nonlinear function. Therefore the presented techniques occupy any high-order convergence. The proposed methods are convenient and self-explanatory, achieve satisfactory results, and also require less and easy computations compared to some current schemes. Experimental results are provided to illustrate the reliability and robustness of the techniques.

Keywords: Approximate inverse, Iterative method, Moore–Penrose.
2010 AMS: 26A18, 65F10, 65F20, 16U50, 15A09.

[1] *Department of Mathematics, Faculty of Science, Yazd University, Yazd, Iran*
***Corresponding author**: e.kokabifar@yahoo.com

Received: 11 April 2020, **Accepted:** 24 June 2020, **Available online:** 30 June 2020

1. Introduction

For every matrix $A \in \mathbb{C}^{m\times n}$, an $m \times n$-matrix of mn complex variables, the Moore-Penrose inverse of A denoted by $A^\dagger$ exists and is a unique matrix for which all the following conditions hold:

$$1.AA^\dagger A = A, \qquad 2.A^\dagger AA^\dagger = A^\dagger, \qquad 3.\left(AA^\dagger\right)^* = AA^\dagger, \qquad 4.\left(A^\dagger A\right)^* = A^\dagger A,$$

in which A^* denotes complex conjugate transpose of the matrix A. The Moore-Penrose pseudo-inverse has a significant role in matrix computation as well as in other fields of computational and applied mathematics. As a result of this, continuous efforts are carried out in order to improve algorithms approximating generalized inverse of a non-square matrix. In particular, some remarkable fields of applications of the concept broadly lie in digital image processing, optimization problems, solution of the system $Ax = b$ and other industrial applications.

Accordingly, finding methods for computing the Moore-Penrose inverse received a meaningful attention of many researchers [1, 2, 3, 4]. For $A \in \mathbb{C}^{n\times n}$, one of the primary and enough efficient and stable computational methods for approximating A^{-1} was introduced by Schulz[5]:

$$X_{k+1} = X_k\left(2I_n - X_kA\right), \qquad k = 0, 1, 2, \ldots. \tag{1.1}$$

where I_n is the identity matrix, and X_0 is an initial value for approximating A^{-1}. This iterative method will quadratically converge to A^{-1}, after enough iterations provided that all eigenvalues of $I_n - X_0A$ are less than 1, i.e., for a matrix norm it holds that $\|I_n - X_0A\| < 1$[6].

Anyway, a family of iterative schemes for approximating the inverse of a square nonsingular matrix proposed by Li et al.[7] as follows:

$$X_{k+1} = X_k \left(pI - \frac{p(p-1)}{2} AX_k + \ldots + (AX_k)^{p-1} \right); \qquad p = 2, 3, \ldots.$$

Each member of the family under the condition $\|I_n - X_0 A\| < 1$ is a convergent iterative method whose order is related to p. It is easy to verify that for $p = 2$ the above relation yields the iterative method (1.1). Furthermore, for $p = 3$ it turns into the following

$$X_{k+1} = X_k \left(3I - 3AX_k + (AX_k)^2 \right), \qquad k = 0, 1, 2, \ldots, \tag{1.2}$$

that is third-order convergence iterative method[7]. Then, Chen and Wang [8] extended the results in [7] for the case of non-square matrices straightforwardly and approximated the Moore-Penrose inverse of them.

In this paper, we are dealing with iterative algorithms that every one of them can find roots of a linear or nonlinear function. Then, it is shown that based on each of which we can establish an iterative technique that is able to approximate inverse of a square matrix and the Moore-Penrose inverse of a non-square one. The presented schemes have different orders of convergence that can arbitrarily increase. Consequently, it yields a matrix inversion method having arbitrary high order convergence.

2. A family of iterative algorithms for solving nonlinear equations

In this section, three well-known iterative algorithms for solving nonlinear equations are reviewed, and a self-explanatory way to extend them is given. Getting along the next member of the family of algorithms, one can achieve a higher order convergence iterative method which of course requires much computational cost. However, for all introduced algorithms, it is needed to compute the first derivate of the given function only. Therefore, it depends on user and the task which iterative technique is more suitable.

Algorithm 2.1. *For a given function $f(x)$ and an initial guess x_0 sufficiently close to a simple root of $f(x)$, the following iterative scheme is known as Newton method and has a quadratic convergence*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 0, 1, 2, \ldots.$$

For a proof of its convergence and more details see, for instance, [9].

Algorithm 2.2. *For every function $f(x)$ and an initial guess x_0 sufficiently close to a simple root of it, the two step Newton method is of the form*

$$x_n^2 = x_n^1 - \frac{f(x_n^1)}{f'(x_n^1)},$$

$$x_{n+1}^1 = x_n^2 - \frac{f(x_n^2)}{f'(x_n^2)}, \qquad n = 0, 1, 2, \ldots,$$

which is of fourth order and proposed by Traub [10].

Algorithm 2.3. *Consider the function $f(x)$ and assume that x_0 is an initial guess sufficiently close to a simple root of $f(x)$. Then*

$$x_n^2 = x_n^1 - \frac{f(x_n^1)}{f'(x_n^1)},$$

$$x_n^3 = x_n^2 - \frac{f(x_n^2)}{f'(x_n^2)},$$

$$x_{n+1}^1 = x_n^3 - \frac{f(x_n^3)}{f'(x_n^3)}, \qquad n = 0, 1, 2, \ldots,$$

is eighth order method for computing a simple root of $f(x)$ [11].

Continuing the above process, it is fairly straightforward to verify that the kth algorithm is as follows. Moreover, pursuing the uncomplicated calculations similar to what is performed in the case of each of the above algorithms will show that the below one has 2^k order convergence.

Algorithm 2.4. *Consider a function $f(x)$ and an initial guess x_0 sufficiently close to a simple root of $f(x)$. Then for every $k \in \mathbb{N}$, where $\mathbb{N}$ denotes the set of natural numbers that is the set of all positive integers, the following iterative scheme converges to the simple zero of the given function*

$$\begin{aligned}
x_n^2 &= x_n^1 - \frac{f(x_n^1)}{f'(x_n^1)}, \\
x_n^3 &= x_n^2 - \frac{f(x_n^2)}{f'(x_n^2)}, \\
&\vdots \\
x_n^k &= x_n^{k-1} - \frac{f(x_n^{k-1})}{f'(x_n^{k-1})}, \\
x_{n+1}^1 &= x_n^k - \frac{f(x_n^k)}{f'(x_n^k)}, \qquad n = 0, 1, 2, \ldots.
\end{aligned}$$

It is worth noting that applying kth $(k = 1, 2, \ldots)$ algorithm when k increases, yields a faster converges but requires much computations, i.e., the more rate of convergence, the more computation cost. Anyway, on the positive side, for every $k = 1, 2, \ldots$ it is needed to compute only first derivative of the given function.

3. A family of iterative methods to approximate the inverse and Moore-Penrose inverse

In the previous section a class of iterative methods for solving $f(x) = 0$ was introduced. Here, it is shown that each member of the family can give us an iterative method to approximate the inverse and the Moore-Penrose inverse. Inheriting its essential properties and corresponding technique, the obtained method for matrix inversion has the same order of convergence. It means that we have iterative methods of arbitrary high order.

To begin with, consider the function $f(x) = \dfrac{1}{x} - a$ and its first derivative that is $f'(x) = \dfrac{-1}{x^2}$. Also, in what follows suppose that $A \in \mathbb{C}^{m \times n}, (m \geq n)$ and I and X_0 are the identity matrix and an initial approximation of desired inverse of the appropriate size, respectively. Then, it is straightforward to verify that

- Algorithm 2.1 easily concludes $x_{n+1} = x_n(2 - ax_n)$ that in its matrix form becomes the following iterative method for computing the inverse or Moore-Penrose inverse of A:

  $$X_{k+1} = X_k(2I_n - AX_k), \qquad k = 0, 1, 2, \ldots,$$

  that is a second-order convergent method.

- Algorithm 2.2 yields

  $$x_{n+1} = x_n\left(4 - 6ax + 4a^2x^2 - a^3x^3\right), \tag{3.1}$$

  which leads to the following iterative method for computing the inverse or the Moore-Penrose inverse of matrix A:

  $$X_{k+1} = X_k\left(4I - 6(AX_k) + 4(AX_k)^2 - (AX_k)^3\right); \qquad k = 0, 1, 2, \ldots. \tag{3.2}$$

  The above iterative method has fourth order convergence.

- Applying Algorithm 2.3 produces

  $$x_{n+1} = x_n\left(8 - 28ax + 56a^2x^2 - 70a^3x^3 + 56a^4x^4 - 28a^5x^5 + 8a^6x^6 - a^7x^7\right),$$

  from which we have the following eight order iterative method for computing the inverse or Moore-Penrose inverse of A:

  $$X_{k+1} = X_k\left(8I - 28(AX_k) + 56(AX_k)^2 - 70(AX_k)^3 + 56(AX_k)^4 - 28(AX_k)^5 + 8(AX_k)^6 - (AX_k)^7\right), \tag{3.3}$$

  where $k = 0, 1, 2, \ldots.$

Certainly, one can continue using Algorithm 2.4 and for greater positive integer k obtain higher order iterative schemes and accordingly higher order iterative methods for matrix inversion. In what follows, in order to avoid unnecessary prolix computations, we restrict ourselves in proving that (3.2) meets all required conditions to be a fourth order iterative method for computing the inverse or the Moore-Penrose inverse of a given matrix. The extension to the remaining iterative schemes is straightforward.

Theorem 3.1. *Let $A \in \mathbb{C}^{m\times n}, (m \geq n)$ and an initial approximation $X_0 \in \mathbb{C}^{n\times m}$ of desired inverse be given. If $\|I - AX_0\| < 1$, then (3.2) converges to the inverse of A and has fourth order convergence.*

Proof. Assume that $\|E_0\| = \|I - AX_0\| < 1$ and similarly for every $k = 1,2,\ldots$ it holds that $\|E_k\| = \|I - AX_k\|$. Then,

$$\begin{aligned} \|E_{k+1}\| &= \|I - AX_{k+1}\| = \left\|I - AX_k\left(4I - 6\left(AX_k\right) + 4(AX_k)^2 - (AX_k)^3\right)\right\| \\ &= \left\|(I - AX_k)^4\right\| = \left\|{E_k}^4\right\| \leq \|E_k\|^4 \leq \|E_{k-1}\|^{16} \leq \ldots \leq \|E_0\|^{4^{k+1}}. \end{aligned}$$

Consequently, $\|E_0\| < 1$ concludes that $\|I - AX_0\| \to 0$ as $k \to \infty$ which implies the convergence of the sequence $X_k, k = 0,1,2,\ldots$.

Besides, to find the order of convergence of (3.2) let Y_k be difference between the desired inverse and X_k. In other words, if the desired inverse is denoted by V, then $Y_k = V - X_k, (k = 0,1,2,\ldots)$. Therefore,

$$AY_{k+1} = I - AX_{k+1} = E_{k+1} = E_k^4 = (AY_k)^4,$$

which for $\alpha = \|A\|^3$ concludes

$$\|Y_{k+1}\| = \left\|Y_k(AY_k)^3\right\| \leq \alpha\|Y_k\|^4,$$

and it means that (3.2) is at least a fourth order convergence. □

Remark 3.2. *It should be noted that for a given matrix A, several initial values for X_0 are introduced. Ben-Israel and Greville [1] proposed one of the most interesting and useful starting matrix as follows*

$$X_0 = \beta A^*, \qquad \beta \in \left(0, \frac{2}{\|A\|_2^2}\right).$$

In addition, $X_0 = \frac{A^T}{\|A\|_1\|A\|_\infty}$ and $X_0 = \mu I, \mu \in \mathbb{R}$ provided that $\|I - \mu A\| < 1$ are some other conceivable starting values. It is also worth mentioning that, for some special matrices, employing particular forms of the stating point may leads to more accurate results. For example,

$$X_0 = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix},$$

is a proper choice when the A is strictly diagonally dominant [12].

Remark 3.3. *The following expression that is clearly equivalent to (3.2), can yield a more accurate approximate inverse or cause rapidly converge to the inverse, especially when are dealing with ill-conditioned matrices*

$$X_{k+1} = X_k\left(4I - AX_k\left(6I - AX_k\left(4I - AX_k\right)\right)\right).$$

Remark 3.4. *For $A \in \mathbb{C}^{m\times n}, (m \leq n)$ there are discussions and calculations completely similar to what already performed in case of $m \geq n$. The only change that must be taken into account is that (3.1) as an algebraic expression can obviously be written as $x_{n+1} = \left(4 - 6xa + 4x^2a^2 - x^3a^3\right)x_n$. Consequently, the iterative scheme (3.2) turns into the following*

$$X_{k+1} = \left(4I - 6\left(X_kA\right) + 4(X_kA)^2 - (X_kA)^3\right)X_k; \qquad k = 0,1,2,\ldots.$$

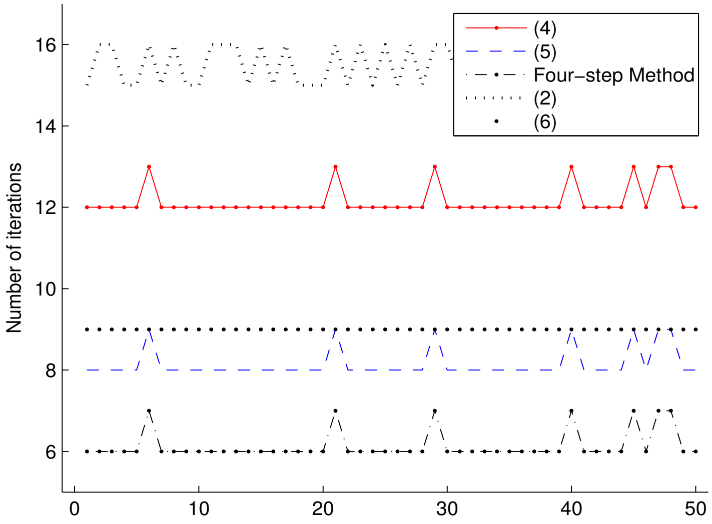

Figure 4.2. Comparison of the number of iterations for 50 matrices

4. Numerical experiments

In this section, the validity and the influence of the proposed methods are examined by a comprehensive example. In the presented experiment iterative methods given by (3.2) and (3.3) are considered. Also, for $k = 4$ Algorithm 2.4, by facile computations analogous to what already discussed in Section 3, will lead to a sixteenth order iterative method for matrix inversion. Denoting it by *Four-step Method*, we also test it in our evaluations. The results obtained by these techniques are compared to the what achieved by the third order method (1.2) and the following seventh-order iterative scheme suggested by Soleymani[13]

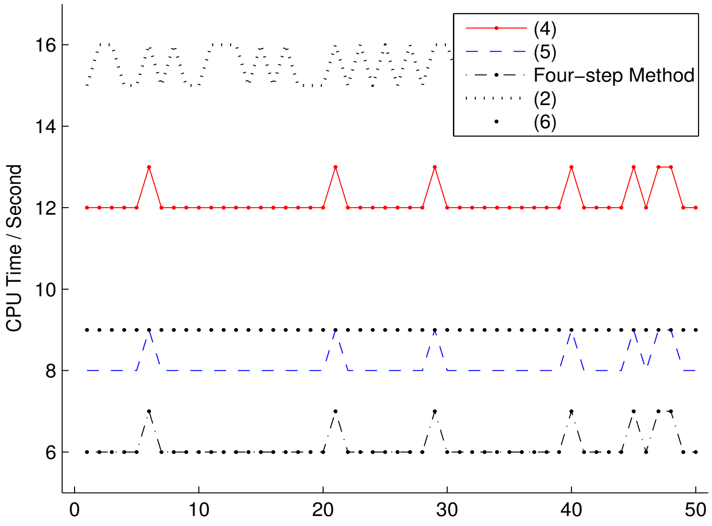

Figure 4.1. Comparison of CPU time for 50 matrices

$$X_{k+1} = \frac{1}{16}X_k \quad [\quad 120I - 3939AX_k + 735(AX_k)^2 - 861(AX_k)^3 + 651(AX_k)^4 - 315(AX_k)^5 \\ + \quad 93(AX_k)^6 - 15(AX_k)^7 + (AX_k)^8], \qquad k = 0, 1, 2, \ldots. \tag{4.1}$$

We examined these five iterative methods employing 50 matrices of size 500 × 450 randomly generated by MATLAB. In our experiments, starting point and the stopping criterion are considered as $X_0 = \frac{1}{2\|A\|_2^2}A^*$ and $\|I - X_kA\| < 10^{-7}$, respectively. The CPU time and number of iterations for these 50 matrices and the five selected iterative methods are shown in Figures 4.1 and 4.2, in the order already mentioned.

5. Conclusion

A family of iterative algorithms to compute either the inverse of an $n \times n$ matrix or the Moore-Penrose inverse of a non-square one was studied. The main point of the illustrated method was the fact that they are arbitrary high-order iterative techniques that can also be used to compute the roots of a both linear and nonlinear function. Finding numerical technique to find other kinds of inverses, particularly those used in real world problems, is the subject of our future research.

References

[1] A. Ben-Israel, T. N. E. Greville, G*eneralized Inverses*, second ed., Springer, 2003.

[2] A. Ben-Israel, A. Charnes, Contributions to the theory of generalized inverses, *SIAM J. App. Maths*, **11** (3) (1963), 667–699.

[3] V. Pan and R. Sehreiber, An improved Newton iteration for the generalized inverse of a matrix, with applications, SIAM J. Sci. Statist. Comput. **12** (1991), 1109–1130.

[4] M. Z. Nashed, X. Chen, Convergence of Newton-like methods for singular operator equations using outer inverses, *Numer. Math.* **66** (1993), 235-257.

[5] G. Schulz, Iterative Berechnung der Reziproken matrix, *Z. Angew. Math. Mech.*, **13** (1933), 57–59.

[6] H. Saberi Najafi, M. Shams Solary, Computational algorithms for computing the inverse of a square matrix, quasi-inverse of a nonsquare matrix and block matrices, *Appl. Math. Comput.*, **183** (2006), 539–550.

[7] W. Li, Z. Li, A family of iterative methods for computing the approximate inverse of a square matrix and inner inverse of a non-square matrix, *Appl. Math. Comput.*, **215** (2010), 3433–3442.

[8] H. Chen, Y. Wang, A family of higher-order convergent iterative methods for computing the Moore–Penrose inverse, *Appl. Math. Comput.*, **218** (2011), 4012–4016.

[9] C. Chun, Iterative methods improving Newton's method by the decomposition method, *Comput. Math. Appl.*, **50** (2005), 1559–1568.

[10] J.F. Traub, *Iterative Methods for Solution of Equations*, Prentice-Hall, Englewood Cliffs, 1964.

[11] M. A. Noor, K. I. Noor, M. Waseem, Higher-order iterative algorithms for solving nonlinear equations, *World Appl. Sci. J.*, **16** (2012), 1657–1663.

[12] L. Grosz, Preconditioning by incomplete block elimination, *Numer. Linear Algebra Appl.*, **7** (2000) 527–541.

[13] F. Soleymani, A Rapid Numerical Algorithm to Compute Matrix Inversion, *Int. J. Maths. Math. Sci.*, (2012), Article ID 134653, 11 pages.