AIMS Mathematics, 8(1): 1127–1138.
DOI: 10.3934/math.2023056
Received: 21 July 2022
Revised: 10 October 2022
Accepted: 11 October 2022
Published: 17 October 2022

http://www.aimspress.com/journal/Math

Research article

An alternating direction power-method for computing the largest singular value and singular vectors of a matrix

Yonghong Duan[1] **and Ruiping Wen**[2,*]

[1] Department of Applied Mathematics, Taiyuan University, Taiyuan 030032, Shanxi, China
[2] Key Laboratory for Engineering and Computing Science, Shanxi Provincial Department of Education, Taiyuan Normal University, Jinzhong 030619, Shanxi, China

* **Correspondence:** Email: wenrp@tynu.edu.cn; Tel: +03512886656.

Abstract: The singular value decomposition (SVD) is an important tool in matrix theory and numerical linear algebra. Research on the efficient numerical algorithms for computing the SVD of a matrix is extensive in the past decades. In this paper, we propose an alternating direction power-method for computing the largest singular value and singular vector of a matrix. The new method is similar to the well-known power method but needs fewer operations in the iterations. Convergence of the new method is proved under suitable conditions. Theoretical analysis and numerical experiments show both that the new method is feasible and is effective than the power method in some cases.

Keywords: singular value decomposition (SVD); power method; alternating direction implicit (ADI); singular value; singular vector; convergence
Mathematics Subject Classification: 65F10, 65F15

1. Introduction

The singular value decomposition (SVD) is a key tool in matrix theory and numerical linear algebra, and plays an important role in many areas of scientific computing and engineering applications, such as least square problem [1], data mining [2], pattern recognition [3], image and signal processing [4,5], statistics, engineering, physics and so on (see [1–6]).

Research on the efficient numerical methods for computing the singular values of a matrix has been a hot topic, many practical algorithms have been proposed for this problem. By using the symmetric QR method to A^TA, Golub and Kahan [7] presented an efficient algorithm named as Golub-Kahan SVD algorithm; Gu and Eisenstat [8] introduced a stable and efficient divide-and-conquer algorithm, called as Divide-and-Conquer algorithm as well as Bisection algorithm for computing the singular value decomposition (SVD) of a lower bidiagonal matrix, see also [1]; Drmac and

Veselic [9] given the superior variant of the Jacobi algorithm and proposed a new one-sided Jacobi SVD algorithm for triangular matrices computed by revealing QR factorizations; many researchers such as Zha [10], Bojanczyk [11], Shirokov [12] and Novaković [13] came up with some methods for the problem of hyperbolic SVD; A Cross-Product Free (CPF) Jacobi-Davidson (JD) type method is proposed to compute a partial generalized singular value decomposition (GSVD) of a large matrix pair (A, B), which is referred to as the CPF-JDGSVD method [14]; many good references for these include [1, 2, 7–9, 15–26] and the references therein for details.

There are important relationships between the SVD of a matrix A and the Schur decompositions of the symmetric matrices A^TA, AA^T and $\begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix}$. These connections to the symmetric eigenproblem allow us to adapt the mathematical and algorithmic developments of the eigenproblem to the singular value problem. So most of the algorithms mentioned above are analogs of algorithms for computing eigenvalues of symmetric matrices. All the algorithms mentioned above except for the Jacobi algorithm, had firstly to reduce A to bi-diagonal form. When the size of the matrix is large, this performance will become very costly. On the other hand, the Jacobi algorithm is rather slowly, though some modification has been added to it (see [9]).

In some applications, such as the compressed sensing as well as the matrix completion problems [3, 27] or computing the 2-norm of a matrix, only a few singular values of a large matrix are required. In these cases, it is obvious that those methods, mentioned the above, for computing the SVD is not very suitable. If only the largest singular value and the singular vectors corresponding to the largest singular value of A is needed, the power method, which is used to approximate a largest eigenpair of an $n \times n$ symmetric matrix A, should be more suitable.

Computing the largest singular value and corresponding singular vectors of a matrix is one of the most important algorithmic tasks underlying many applications including low-rank approximation, PCA, spectral clustering, dimensionality reduction, matrix completion and topic modeling. This paper consider the problem of computing the largest singular value and singular vectors corresponding to the largest singular value of a matrix. We propose an alternating direction method, a fast general purpose method for computing the largest singular vectors of a matrix when the target matrix can only be accessed through inaccurate matrix-vector products. In the other words, the proposed method is analogous to the well-known power method, but has much better numerical behaviour than the power method. Numerical experiments show that the new method is more effective than the power method in some cases.

The rest of the paper is organized as follows. Section 2 contains some notations and some general results that are used in subsequent sections. In Section 3 we propose the alternating direction power-method in detail and give its convergence analysis. In Section 4, we use some experiments to show the effectiveness of the new method. Finally, we end the paper with a concluding remark in Section 5.

2. Preliminaries

The following are some notations and definitions we will use later.

We use $\mathbb{R}^{m\times n}$ to denote the set of all real $m \times n$ matrices, and $\mathbb{R}^n$ the set of real $n \times 1$ vectors. The symbol I denotes the $n \times n$ identity matrix. For a vector $x \in \mathbb{R}^n$, $\|x\|_2$ denotes the 2-norm of x. For a matrix $A \in \mathbb{R}^{n\times n}$, A^T is used to express the transpose of A, rank(A) is equal to the rank of a matrix A,

$\|A\|_2$ denotes the 2-norm of A and the Frobenius norm by $\|A\|_F$ is the maximum absolute value of the matrix entries of a matrix A. $\mathrm{diag}(a_1, a_2, \ldots, a_n)$ represents the diagonal matrix with diagonal elements $a_1, a_2, \ldots, a_n$.

If $A \in \mathbb{R}^{m\times n}$, then there exist two orthogonal matrices

$$U = [u_1, u_2, \ldots, u_m] \in \mathbb{R}^{m\times m} \text{ and } V = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^{n\times n}$$

such that

$$U^T AV = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix}, \tag{2.1}$$

where $\Sigma_r = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0, r = \mathrm{rank}(A) \leqslant \min\{m, n\}$. The σ_i are the singular values of A and the vectors u_i and v_i are the ith left singular vector and the ith right singular vector respectively. And we have the SVD expansion

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T.$$

This is the well-known singular value decomposition (SVD) theorem [2].

Lemma 2.1. (see Lemma 1.7 and Theorem 3.3 of [1]) Let $A \in \mathbb{R}^{m\times n}$. Then $\|A\|_2 = \|A^T\|_2 = \sqrt{\|AA^T\|_2} = \sigma_1$, where σ_1 is the largest singular value of A.

Lemma 2.2. (see Theorem 3.3 of [1]) Let $A \in \mathbb{R}^{m\times n}$ and σ_i, u_i, v_i, $i = 1, 2, \ldots, r$ be the singular values and the corresponding singular vectors of A respectively. Then

$$AA^T u_i = \sigma_i^2 u_i, \quad A^T A v_i = \sigma_i^2 v_i, \quad i = 1, 2, \ldots, r.$$

Lemma 2.3. (refer to Section 2.4 of [2] or Theorem 3.3 of [1])) Assume the matrix $A \in \mathbb{R}^{m\times n}$ has rank $r > k$ and the SVD of A be (2.1). The matrix approximation problem $\min\limits_{\mathrm{rank}(Z)=k} \|A - Z\|_F$ has the solution

$$Z = A_k = U_k \Sigma_k V_k^T,$$

where $U_k = (u_1, \ldots, u_k)$, $V_k = (v_1, \ldots, v_k)$ and $\Sigma_k = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$.

Let $A \in \mathbb{R}^{n\times n}$. The power method for computing the module largest eigenvalue of A is as follows (see as Algorithm 4.1 of [1]).

Power method:
(1) Choose an initial vector $x_0 \in \mathbb{R}^n$. For $k = 0, 1, \ldots$ until convergence;
(2) Compute $y_{k+1} = Ax_k$;
(3) Compute $x_{k+1} = y_{k+1}/\|y_{k+1}\|_2$;
(4) Compute $\lambda_{k+1} = x_{k+1}^T A x_{k+1}$;
(5) Set $k = k + 1$ and go to (2).

The power method is very simple and easy to implement and is applied in many applications, for example, for the PCA problem (see [28]).

To compute the largest singular value and the corresponding singular vectors of A, we can apply the power method to AA^T or A^TA, without actually computing AA^T or A^TA.

Power method (for largest singular value):
(1) Choose an initial vector $u_0 \in \mathbb{R}^m$ and $v_0 \in \mathbb{R}^n$. For $k = 0, 1, \ldots$ until convergence;
(2) Compute $y_{k+1} = A(A^T u_k)$, $z_{k+1} = A^T(Av_k)$;
(3) Compute $u_{k+1} = y_{k+1}/\|y_{k+1}\|_2$, $v_{k+1} = z_{k+1}/\|z_{k+1}\|_2$;
(4) Compute $\lambda_{k+1} = u_{k+1}^T A u_{k+1}$ and $\sigma_{k+1} = \sqrt{\lambda_{k+1}}$;
(5) Set $k = k + 1$ and go to (2).

However, the power method will cost extra operations if the two singular vectors are needed and, in some cases, it converges very slowly. So, in the next section, we will propose a new iteration method for computing the largest singular value and the corresponding singular vectors of a matrix, which is similar to the power method but needs fewer operations in the iterations.

3. An alternating direction power-method

In this section, we will introduce an alternating direction power iteration method. The new method is based on an important property of the SVD.

From Lemma 2.3, it is known that the largest singular value σ_1 and the corresponding singular vectors u_1, v_1 of A satisfy the following condition

$$\|A - \sigma_1 u_1 v_1^T\|_F = \min_{u,v} \|A - uv^T\|_F,$$

where $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$. Thus the problem of computing σ_1, u_1 and v_1 is equivalent to solve the optimization problem

$$\min_{u,v} \|A - uv^T\|_F.$$

Let $f(u, v) = \frac{1}{2}\|A - uv^T\|_F^2$. For the sake of simplicity, we will solve the equivalent optimization problem

$$\min_{u,v} f(u, v) = \min_{u,v} \frac{1}{2}\|A - uv^T\|_F^2, \tag{3.1}$$

where $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$. However, the problem (3.1) is difficult to solve since it is not convex for u and v. Fortunately, it is convex for u individually and so is v, so we can use the alternating direction method to solve it. Alternating minimization is widely used in optimization problems due to its simplicity, low memory and flexibility (see [20, 29]). In the following we apply an alternating method to solve the unconstrained optimization problem (3.1).

Suppose v_k was known, then (3.1) can be reduced to the unconstrained optimization problem

$$\min_{u} f(u, v_k) = \min_{u} \frac{1}{2}\|A - uv_k^T\|_F^2. \tag{3.2}$$

The (3.2) can be solved by many efficient methods, such as steepest decent method, Newton method, conjugate gradient (CG) method and so on (see [29]). Because Newton method is simple and converges fast, we will choose to use the Newton method. By direct calculation, we get

$$\nabla_u f = -(A - uv^T)v, \quad \Delta_u f = \|v\|_2^2 I.$$

Then applying Newton method, we get

$$\begin{aligned} u_{k+1} &= u_k - (\Delta_u f)^{-1}\nabla_u f \\ &= u_k + \frac{1}{\|v_k\|_2^2}(A - u_k v_k^T)v_k \\ &= \frac{1}{\|v_k\|_2^2}Av_k. \end{aligned}$$

Alternatively, when u_{k+1} is known, the problem (3.1) can be reduced to

$$\min_v f(u_{k+1}, v) = \min_v \frac{1}{2}\|A - u_{k+1}v^T\|_F^2. \tag{3.3}$$

Also,

$$\nabla_v f = -(A - uv^T)^T u, \quad \Delta_v f = \|u\|_2^2 I,$$

it is obtained that by applying Newton method

$$\begin{aligned} v_{k+1} &= v_k - (\Delta_v f)^{-1}\nabla_v f \\ &= v_k + \frac{1}{\|u_{k+1}\|_2^2}(A - u_{k+1}v_k^T)^T u_{k+1} \\ &= \frac{1}{\|u_{k+1}\|_2^2}A^T u_{k+1}. \end{aligned}$$

Solving (3.2) and (3.3) to high accuracy is both computationally expensive and of limited value if u_k and v_k are far from stationary points. So, in the following, we apply the two iterations alternately. Thus the alternating directional Newton method for solving (3.1) is

$$\begin{cases} u_{k+1} = & \frac{1}{\|v_k\|_2^2}Av_k \\ v_{k+1} = & \frac{1}{\|u_{k+1}\|_2^2}A^T u_{k+1} \end{cases}, \quad k = 0, 1, \ldots, \tag{3.4}$$

where $u_0 \in \mathbb{R}^m$ and $v_0 \in \mathbb{R}^n$ are both initial guesses. At each iteration, only two matrix-vector multiplications are required and the operation costs are about $4mn$, which is less than that of the power method.

Next, the convergence analysis of (3.4) would be provided.

Theorem 3.1 Let $A \in \mathbb{R}^{m\times n}$ and σ_1, u, v be the largest singular value and the corresponding singular vectors of A respectively. If $u_0 \in \mathbb{R}^m$ and $v_0 \in \mathbb{R}^n$ are both initial guesses such that the projections on u and v are not zero, then the iteration (3.4) is convergent with

$$\lim_{k\to\infty}\frac{u_k}{\|u_k\|_2} = u, \quad \lim_{k\to\infty}\frac{v_k}{\|v_k\|_2} = v,$$

and

$$\lim_{k\to\infty}\|u_k\|_2 \cdot \|v_k\|_2 = \sigma_1.$$

Proof. From (3.4) we can deduce that

$$u_{k+1} = \frac{1}{\|v_k\|_2^2} A v_k = \frac{\|u_k\|_2^2}{\|A^T u_k\|_2^2} A A^T u_k,$$

$$v_{k+1} = \frac{1}{\|u_{k+1}\|_2^2} A^T u_{k+1} = \frac{\|v_k\|_2^2}{\|A v_k\|_2^2} A^T A v_k.$$

As in the proof of the power method (see as [2]), if the projections of u_0 on u, and v_0 on v are not zero, then we have

$$\lim_{k\to\infty} \frac{u_k}{\|u_k\|_2} = u, \quad \lim_{k\to\infty} \frac{v_k}{\|v_k\|_2} = v.$$

On the other hand, we have

$$\begin{aligned} \|u_{k+1}\|_2 \cdot \|v_{k+1}\|_2 =& \frac{1}{\|v_k\|_2^2} \|A v_k\|_2 \frac{\|v_k\|_2^2}{\|A v_k\|_2^2} \|A A^T u_k\|_2 \\ =& \frac{1}{\|A v_k\|_2} \|A A^T u_k\|_2 \\ \leq& \|A^T\|_2 \\ =& \sigma_1. \end{aligned}$$

Thus the sequence $\{\|u_k\|_2 \cdot \|v_k\|_2\}$ is bounded from the above. By

$$\|u_{k+1}\|_2 \cdot \|v_{k+1}\|_2 = \frac{1}{\|v_k\|_2^2} \|A v_k\|_2 \frac{1}{\|u_{k+1}\|_2^2} \|A^T u_{k+1}\|_2,$$

we can conclude that when $k \to \infty$

$$\begin{aligned} \|u_{k+1}\|_2^2 \cdot \|v_{k+1}\|_2 \cdot \|v_k\|_2 =& \|A \frac{1}{\|v_k\|_2} v_k\|_2 \cdot \|A^T \frac{1}{\|u_{k+1}\|_2} u_{k+1}\|_2 \\ \to& \|A v\|_2 \cdot \|A^T u\|_2 \\ =& \sigma_1^2. \end{aligned}$$

Therefore,

$$\lim_{k\to\infty} \|u_k\|_2 \cdot \|v_k\|_2 = \sigma_1.$$

□

We now propose the alternating direction power-method with the discussions above for computing the largest singular value and corresponding singular vectors of a matrix as follows.

Alternating Direction Power-Method (ADPM):
(1) Choose initial vectors $v_0 \in \mathbb{R}^n$. For $k = 0, 1, \ldots$ until convergence;
(2) Compute $u_{k+1} = A v_k / \|v_k\|_2^2$;
(3) Compute $v_{k+1} = A^T u_{k+1} / \|u_{k+1}\|_2^2$;
(4) Compute $\mu_{k+1} = \|u_{k+1}\|_2 \cdot \|v_{k+1}\|_2$;
(5) Set $k = k + 1$ and go to (2).

4. Numerical experiments

Here, we use several examples to show the effectiveness of the alternating direction power-method (ADPM). We compare ADPM with the power method (PM) and present numerical results in terms of the numbers of iterations (IT), CPU time (CPU) in seconds and the residue (RES), where the measurement method of CPU time in seconds is uniformly averages over multiple runs by embed MATLAB functions `tic/toc` at each iteration step and

$$\text{RES} = \text{abs}\left(\|u_{k+1}\|_2 \cdot \|v_{k+1}\|_2 - \|u_k\|_2 \cdot \|v_k\|_2\right).$$

The initial vectors u_0 and v_0 are chosen randomly by the MATLAB statements $u_0 = rand(m, 1)$ and $v_0 = rand(n, 1)$. In our implementations all iterations are performed in MATLAB (R2016a) on the same workstation with an Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz that has 16GB memory and 32-bit operating system, and are terminated when the current iterate satisfies RES$< e^{-12}$ or the number of iterations is more than 9000, which is denoted by '-'.

Experiment 4.1. In the first experiment, we generate random matrices with uniformly distributed elements by the MATLAB statement

$$\texttt{A = rand(m, n)}.$$

For different sizes of m and n, we apply the power method and the alternating direction power-method with numerical results reported in Table 1.

Table 1. Numerical results of Experiment 4.1.

m	n	Method	IT	CPU	RES	RATIO(%)
500	100	PM	7	0.001340	1.4211e-13	
		ADPM	4	0.000471	1.4172e-15	35.15
500	200	PM	7	0.002753	5.6843e-14	
		ADPM	4	0.000641	2.8422e-14	23.28
1000	200	PM	7	0.015660	2.8422e-14	
		ADPM	4	0.003914	1.7053e-13	24.99
1000	500	PM	6	0.017024	2.8422e-13	
		ADPM	3	0.004934	2.8422e-13	28.98
2000	500	PM	6	0.030742	5.6843e-14	
		ADPM	3	0.016660	1.1369e-13	54.19
2000	1000	PM	6	0.060125	4.5475e-13	
		ADPM	3	0.014466	1.1369e-13	24.05

Experiment 4.2. In this experiment, we generate random matrices with normally distributed elements by

$$\texttt{A = randn(m, n)}.$$

For different sizes of m and n, we apply the power method and the alternating direction power-method to A.

Numerical results are reported in Table 2.

Table 2. Numerical results of Experiment 4.2.

m	n	Method	IT	CPU	RES	RATIO(%)
500	100	PM	711	0.084958	9.2371e-13	
		ADPM	372	0.046376	9.6634e-13	54.59
500	200	PM	652	0.084715	8.8818e-13	
		ADPM	449	0.058401	9.3081e-13	68.94
1000	200	PM	1124	0.159432	9.9476e-13	
		ADPM	700	0.121118	9.7344e-13	75.96
1000	500	PM	1288	1.122289	9.8055e-13	
		ADPM	782	0.378765	9.8765e-13	33.75
2000	500	PM	1744	3.627609	9.6634e-13	
		ADPM	961	1.221796	9.0949e-13	33.68
2000	1000	PM	4364	20.032671	9.9476e-13	
		ADPM	2377	5.696365	9.0949e-13	28.42

Experiment 4.3. In this experiment, we use some test matrices with size $n \times n$ from the university of Florida sparse matrix collection [30]. Numerical results are reported in Table 3.

Table 3. Numerical results of Experiment 4.3.

Matrix	size	Method	IT	CPU	RES	RATIO(%)
lshp1009	1009	PM	1488	0.064107	9.9298e-13	
		ADPM	745	0.019718	9.9654e-13	30.76
dwt_1005	1035	PM	40	0.005109	7.2120e-13	
		ADPM	35	0.003844	6.2172e-13	75.24
bcsstk13	2003	PM	1519	0.511403	9.7877e-13	
		ADPM	842	0.174025	9.9920e-13	34.03
dwt_2680	2680	PM	514	0.070507	1.4172e-09	
		ADPM	239	0.046361	9.7167e-13	65.75
rw5151	5151	PM	1006	0.128023	9.8632e-13	
		ADPM	590	0.057257	9.7056e-13	44.72
g7jac040	11790	PM	26	0.038169	0	
		ADPM	17	0.012170	0	31.88
epb1	14734	PM	-	-	-	-
		ADPM	6132	1.541585	9.9926e-13	-

In particular, compared with the cost of the power method, we can find that the cost of the alternating direction power-method is discounted, up to 23.28%. The "RATIO", defined in the following, in the Tables 1–3 can show this effectiveness.

$$\text{RATIO} = \frac{\text{the CPU of the ADPM}}{\text{the CPU of the PM}} \times 100\%.$$

In order to show numerical behave of two methods, the cost curves of the methods are clearly given,

which are shown in Figures 1–3.

From Tables 1–3, we can conclude that ADPM needs fewer iterations and less CPU time than the power method. So it is feasible and is effective in some cases.

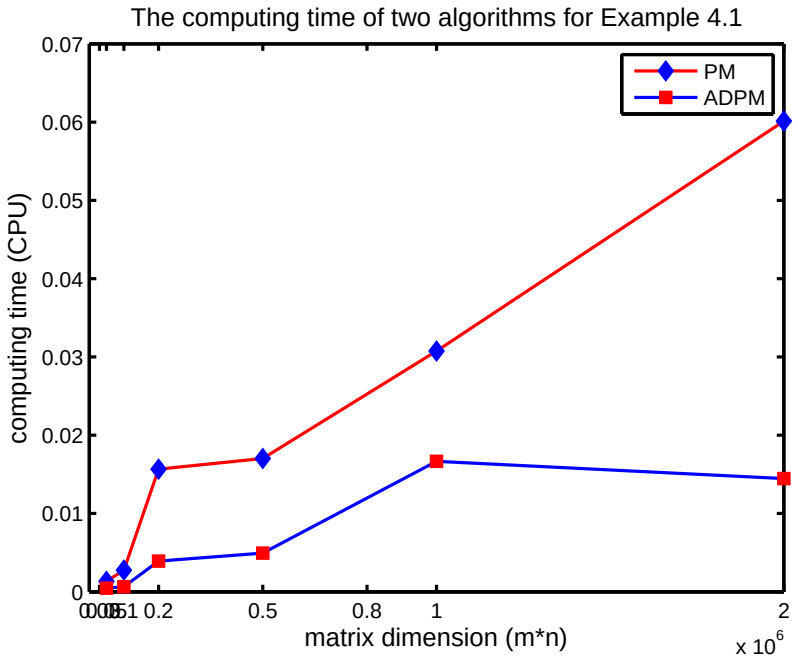

Figure 1. Comparison curve of the PM and ADPM methods for Example 4.1.

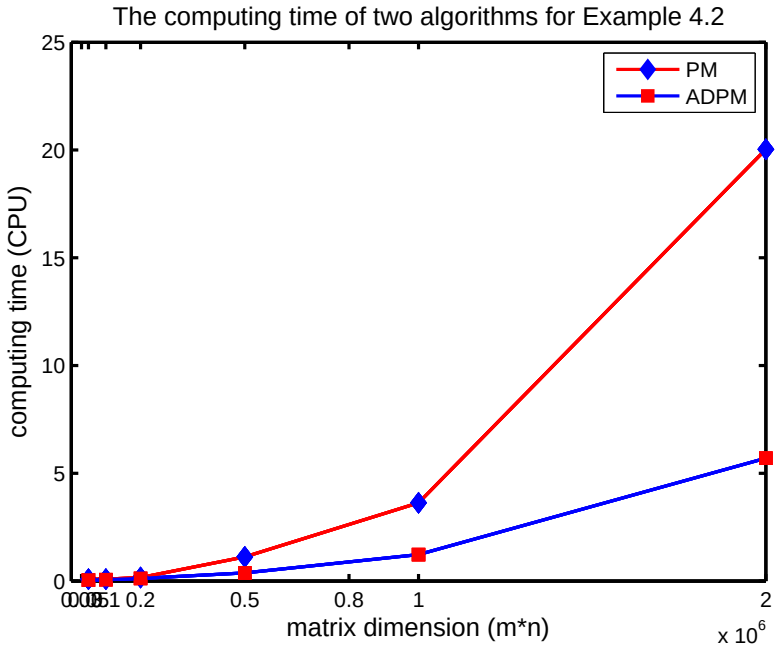

Figure 2. Comparison curve of the PM and ADPM methods for Example 4.2.

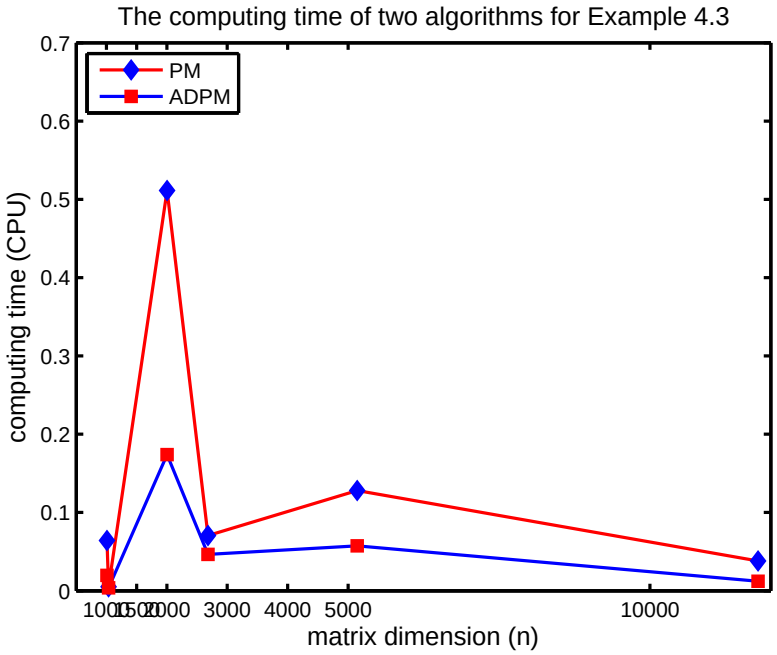

Figure 3. Comparison curve of the PM and ADPM methods for Example 4.3.

5. Conclusions

In this study, we have proposed an alternating direction power-method for computing the largest singular value and singular vector of a matrix, which is analogs to the power method but needs fewer operations in the iterations since using the technique of alternating. Convergence of the alternating direction power-method is proved under suitable conditions. Numerical experiments have shown that the alternating direction power-method is feasible and more effective than the power method in some cases.

Acknowledgments

The authors are very much indebted to the anonymous referees for their helpful comments and suggestions which greatly improved the original manuscript of this paper. The authors are so thankful for the support from the NSF of Shanxi Province (201901D211423) and the scientific research project of Taiyuan University, China (21TYKY02).

Conflict of interest

The authors declare that they have no conflict of interests.

References

1. J. Demmel, *Applied numerical linear algebra*, Philadelphia: SIAM, 1997. https://doi.org/10.1137/1.9781611971446

2. G. H. Golub, C. F. Van Loan, *Matrix computations*, 4 Eds., Baltimore and London: The Johns Hopkins University Press, 2013.

3. J. F. Cai, E. J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optim.,* **20** (2010), 1956–1982. https://doi.org/10.1137/080738970

4. L. L. Scharf, The SVD and reduced rank signal processing, *Signal Process.,* **25** (1991), 113–133. https://doi.org/10.1016/0165-1684(91)90058-Q

5. B. D. Moor, The singular value decomposition and long and short spaces of noisy matrices, *IEEE Trans. Signal Proc.,* **41** (1993), 2826–2838. https://doi.org/10.1109/78.236505

6. M. V. Kulikova, Hyperbolic SVD-based Kalman filtering for Chandrasekhar recursion, *IET Control Theory Appl.,* **13** (2019), 1525. https://doi.org/10.1049/iet-cta.2018.5864

7. G. H. Golub, W. Kahan, Calculating the singular values and the pesudo-inverse of a matrix, *SIAM J. Numer. Anal.,* **2** (1965), 205–224. https://doi.org/10.1137/0702016

8. M. Gu, S. C. Eisenstat, A divide-and-conquer algorithm for the bidiagonal SVD, *SIAM J. Matrix Anal. Appl.,* **16** (1995), 79–92. https://doi.org/10.1137/S0895479892242232

9. Z. Drmac, K. Veselic, New fast and accurate Jacobi SVD algorithm, *SIAM J. Matrix Anal. Appl.,* **29** (2008), 1322–1362. https://doi.org/10.1137/050639193

10. H. Zha, A note on the existence of the hyperbolic singular value decomposition, *Linear Algebra Appl.,* **240** (1996), 199–205. https://doi.org/10.1016/0024-3795(94)00197-9

11. A. W. Bojanczyk, An implicit Jacobi-like method for computing generalized hyperbolic SVD, *Linear Algebra Appl.,* **358** (2003), 293–307. https://doi.org/10.1016/S0024-3795(02)00394-4

12. D. S. Shirokov, A note on the hyperbolic singular value decomposition without hyperexchange matrices, *J. Comput. Appl. Math.,* 2021, 1–8.

13. V. Novaković, S. Singer, A GPU-based hyperbolic SVD algorithm, *BIT Numer. Math.,* **51** (2011), 1009–1030. https://doi.org/10.1007/s10543-011-0333-5

14. J. Huang, Z. Jia, A cross-product free Jacobi-Davidson type method for computing a partial generalized singular value decomposition (GSVD) of a large matrix pair, 2020, 1–14.

15. J. Demmel, P. Koev, Accurate SVDs of weakly diagonally dominant M-matrices, *Numer. Math.,* **98** (2004), 99–104. https://doi.org/10.1007/s00211-004-0527-8

16. J. Demmel, P. Koev, Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials, *Linear Algebra Appl.,* **417** (2006), 382–396. https://doi.org/10.1016/j.laa.2005.09.014

17. J. Demmel, W. Kahan, Accurate singular values of bidiagonal matrices, *SIAM J. Stat. Comp.,* **11** (1990), 873–912. https://doi.org/10.1137/0911052

18. J. Demmel, Accurate singular value decompositions of structured matrices, *SIAM J. Matrix Anal. Appl.,* **21** (1999), 562–580. https://doi.org/10.1137/S0895479897328716

19. F. M. Dopico, J. Moro, A note on multiplicative backward errors of accurate SVD algorithms, *SIAM J. Matrix Anal. Appl.,* **25** (2004), 1021–1031. https://doi.org/10.1137/S0895479803427005

20. R. Escalante, M. Raydan, *Alternating projection methods*, Philadelphia: SIAM, 2007.

21. V. Fernando, B. N. Parlett, Accurate singular values and differential qd algorithms, *Numer. Math.,* **67** (1994), 191–229. https://doi.org/10.1007/s002110050024

22. B. Grosser, B. Lang, An $O(n^2)$ algorithm for the bidiagonal SVD, *Linear Algebra Appl.,* **358** (2003), 45–70. https://doi.org/10.1016/S0024-3795(01)00398-6

23. R. A. Horn, C. R. Johnson, *Matrix analysis*, Cambridge: Cambridge University Press, 1985. https://doi.org/10.1017/CBO9780511810817

24. N. J. Higham, QR factorization with complete pivoting and accurate computation of the SVD, *Linear Algebra Appl.,* **309** (2000), 153–174. https://doi.org/10.1016/S0024-3795(99)00230-X

25. P. Koev, Accurate eigenvalues and SVDs of totally nonnegative matrices, *SIAM J. Matrix Anal. Appl.,* **27** (2005), 1–23. https://doi.org/10.1137/S0895479803438225

26. Q. Ye, Computing singular values of diagonally dominant matrices to high relative accuracy, *Math. Comput.,* **77** (2008), 2195–2230. https://doi.org/10.1090/S0025-5718-08-02112-1

27. J. Blanchard, J. Tanner, K. Wei, *CGIHT: Conjugate grandient iterative hard thresholding for compressed sensing and matrix completion*, University of Oxford, 2014. https://doi.org/10.1093/imaiai/iav011

28. M. Hardt, E. Price, *The noisy power method: A meta algorithm with applications,* International Conference on Neural Information Processing Systems MIT Press, 2014.

29. J. Nocedal, S. Wright, *Numerical optimization*, Berlin: Springer, 2006.

30. T. Davis, Y. Hu, The University of Florida sparse matrix collection, *ACM Trans. Math. Software,* **38** (2011), 1–25. https://doi.org/10.1145/2049662.2049663

AIMS Press