AIMS Mathematics, 11(11): 39607–39620.
DOI: 10.3934/math.20261525
Received: 13 October 2026
Revised: 11 November 2026
Accepted: 19 November 2026
Published: 26 November 2026

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

*Research article*

# Existence criteria of the MPBT-inverse and its randomized algorithms

**Sanzhang Xu**[1,*], **Zhengyang Shan**[1], **and Can Zha**[1,2]

[1] Faculty of Mathematics and Physics, Huai'an University, Huaian, 223003, China
[2] School of Mathematical Sciences, Suzhou University of Science and Technology, Jiangsu 215009, China

* **Correspondence:** Email: szxu@hau.edu.cn; Tel: +86-159-5172-7351.

**Abstract:** For a complex square matrix, we proved that every MPBT (MPBT is the abbreviation of Moore-Penrose, Baksalary and Trenkler) invertible matrix was Bott-Duffin $(A^{\dagger}A_1^{\diamond}, A_1^{\diamond}A^{\dagger})$-invertible. We constructed a Gradient Neural Network model $GNNSYS(A, A_1^{\diamond}A^{\dagger}, A, A^{\dagger}A_1^{\diamond})$ for the system $AX = A_1^{\diamond}A^{\dagger}$ and $XA = A^{\dagger}A_1^{\diamond}$, where $A_1^{\diamond} = AA^{\diamond}A$ and the above system satisfied the Mitra compatibility conditions. The state matrix $V(t)$ in the above system converged to the MPBT-inverse. We verified the main theorem by using numerical examples.

**Keywords:** Moore-Penrose inverse; MPBT-inverse; Bott-Duffin(P,Q)-inverse; gradient neural network model; randomized algorithm
**Mathematics Subject Classification:** 15A90, 68W20, 65F10

## 1. Introduction

The Hartwig-Spindelböck decomposition was introduced in [6], which says that $A \in \mathbb{C}^{n\times n}$ with $\mathrm{ind}(A) = r$ can be written as

$$A = U\begin{bmatrix}\Sigma K & \Sigma L\\ O & O\end{bmatrix}U^{*} \tag{1.1}$$

where $U \in \mathbb{C}^{n\times n}$ is unitary, $\Sigma \in \mathbb{C}^{r\times r}$ is invertible, and $K \in \mathbb{C}^{r\times r}$ and $L \in \mathbb{C}^{r\times(n-r)}$. All generalized inverses computation problems studied in this paper can be related $AX = C$ and $XB = D$, where $A \in \mathbb{C}^{p\times q}$, $C \in \mathbb{C}^{p\times s}$, $B \in \mathbb{C}^{s\times t}$, $D \in \mathbb{C}^{q\times t}$, for example, the MPBT inverse in [12, Theorem 3.1]. C.R. Rao and S.K. Mitra [10] gave the compatibility criterion for the existence of a common solution to this system. Compatibility criterion for the existence of a common solution to this system is called the Mitra compatibility condition [10]. Let $A \in \mathbb{C}^{m\times n}$. If a matrix $X \in \mathbb{C}^{n\times m}$ satisfies $AXA = A$, $XAX = X$, $(AX)^{*} = AX$ and $(XA)^{*} = XA$, then $X$ is called the Moore-Penrose inverse of $A$ [8] and denoted by $X = A^{\dagger}$. The BT-inverse (BT-inverse is the abbreviation of Baksalary and Trenkler inverse) of $A$

is the Moore-Penrose of $A^2A^\dagger$ [1, Definition 1]. In view of (1.1), one can get that the expression of the BT-inverse of $A$ is $A^\diamond = U\begin{bmatrix} (\Sigma K)^\dagger & 0 \\ 0 & 0 \end{bmatrix}U^*$. Let $A_1^\diamond = AA^\diamond A$, and the matrix $A^\dagger A_1^\diamond A^\dagger$ is called the MPBT inverse of $A$ [12, Theorem 3.1], an abbreviation of Moore-Penrose Baksalary and Trenkler inverse. In [3, Definition 3.2], one can see the definition of the Bott-Duffin $(P, Q)$-inverse of $A$. Let $A, P, Q \in \mathbb{C}^{n\times n}$, then $Y \in \mathbb{C}^{n\times n}$ is called the Bott-Duffin $(P, Q)$-inverse of $A$ if $Y$ satisfies $PY = Y, YQ = Y, YAP = P, QAY = Q$.

The gradient neural network (GNN) is a continuous-time dynamical neural network model constructed based on the gradient descent principle [2]. As a typical class of continuous-time neural dynamics solvers, the neural network method has been widely applied in matrix inversion and generalized inverse computation: P.S. Stojanović et al. [11] systematically discussed the construction and performance of the Zeroing Neural Network (ZNN) model based on hyperpower iteration for matrix inversion. Such research lays an important foundation for extending the gradient neural network systems (GNNSYS) model to the computation of the MPBT inverse in this paper. The core idea is to construct a dynamical system for a given mathematical problem, such that the state matrix $V(t)$ of the system gradually converges to the solution of the problem as time $t$ evolves. Unlike traditional static numerical algorithms, GNN models the solving process itself as a continuously evolving trajectory, offering good parallelism and implementation potential.

The construction process of GNN can be divided into the following three steps [2]:

**Step 1**: Construction of the error function. For the error matrix $E(t)$, the corresponding error function is constructed as

$$\varepsilon(V(t)) = \frac{1}{2} \parallel E(t) \parallel_F^2,$$

where the symbol $\|\cdot\|_F$ is the Frobenius norm.

**Step 2**: Establishment of the dynamical equation:

$$\dot{V}(t) = -\gamma\mathcal{F}\left(\frac{\partial\varepsilon(V(t))}{\partial V}\right),$$

where $\gamma > 0$ is a scaling parameter controlling the evolution rate of the system, and $\mathcal{F}(\cdot)$ is an element-wise defined activation function.

**Step 3**: For $AX = C$ and $XB = D$, J. Dakić and M.D. Petković constructed the GNNSYS model [2]. The dynamical equation of the GNNSYS model is obtained.

$$\dot{V}(t) = \gamma\mathcal{F}\left(A^T(C - AV(t)) + (D - V(t)B)B^T\right). \tag{1.2}$$

The activation function $\mathcal{F}(\cdot)$ is adopted in [2]: Linear activation function, Power-Sigmoid activation function, and smooth power-Sigmoid activation function. Let $p$ and $q$ are integers such that $p$ is odd, $p > 1$ and $q > 2$. Then, the Power-Sigmoid activation function is

$$f_{ps}(x, p, q) = \begin{cases} \frac{1+\exp(-q)}{1-\exp(-q)} \cdot \frac{1-\exp(-qx)}{1+\exp(-qx)} & \text{if } |x| < 1, \\ x^p & \text{if } |x| \geqslant 1. \end{cases}$$

The smooth power-Sigmoid activation function is

$$f_{sps}(x, p, q) = \frac{1 + \exp(-q)}{1 - \exp(-q)} \cdot \frac{1 - \exp(-qx)}{1 + \exp(-qx)} + \frac{1}{2}x^p.$$

The choice of these three activation functions makes a substantial difference in several scenarios. It is important when the target mapping is inherently nonlinear, when bounded outputs between 0 and 1 are needed, when stable gradients for optimization are prioritized, and when adjustable saturation characteristics are required to fit input features. Relative to the other two, linear activation offers constant gradients, zero risk of gradient saturation and minimal computation. It is suitable for unbounded regression problems but lacks nonlinear capacity. Power-sigmoid surpasses linear activation by introducing tunable nonlinearity and bounded outputs; meanwhile, it has a simpler expression than smooth power-sigmoid and allows flexible shaping of the activation curve via its power parameter. Smooth power-sigmoid inherits the adjustable nonlinearity of power-sigmoid and further achieves full smoothness everywhere. This guarantees continuous gradients at all points, which prevents unstable updates that may arise from non-differentiable points in ordinary power-sigmoid during training.

## 2. Existence criteria of the MPBT-inverse

With the rapid increase in matrix dimensions in large-scale engineering computations, seeking efficient computational paradigms for generalized inverses has become one of the core topics in numerical linear algebra. This chapter systematically establishes computational schemes for the BT-inverse and MPBT inverse based on randomized singular value decomposition (RSVD), which complements the GNNSYS method: GNNSYS belongs to the continuous-time dynamical system approach, with convergence guaranteed by Lyapunov stability; RSVD belongs to the static matrix decomposition approach, compressing a high-dimensional matrix into a low-dimensional subspace via Gaussian random projections, trading a controllable loss of accuracy for a significant improvement in computational efficiency. By [1, Definition 1] and [12, Theorem 3.1], the direct computations of the BT-inverse and MPBT inverse are respectively

$$A^{\diamond} = (A^2A^{\dagger})^{\dagger},\ A^{\dagger,\diamond} = A^{\dagger}AA^{\diamond}$$

both of which rely on multiple Moore-Penrose inverse operations. Many practical matrices have low numerical rank, where a few principal singular values dominate most of the energy. RSVD uses a Gaussian random projection matrix $\Omega \in \mathbb{C}^{n\times(k+p)}$ to compress $A$ into a low-dimensional subspace and then performs an exact SVD, yielding a significant efficiency advantage when $k \ll n$. This idea is consistent with the classical subspace approximation theory in matrix analysis [5], the difference being that RSVD replaces deterministic singular value truncation with randomized sampling, thereby obtaining substantial computational acceleration for large-scale problems. This section starts from the two idempotents $A^{\dagger}A_1^{\diamond}$ and $A_1^{\diamond}A^{\dagger}$. In the following theorem, we will prove that a MPBT invertible matrix is Bott-Duffin $(P, Q)$-invertible.

**Theorem 2.1.** *Let $A \in \mathbb{C}^{n\times n}$. Then, a MPBT invertible matrix is Bott-Duffin $(P, Q)$-invertible, where $P = A^{\dagger}A_1^{\diamond}$ and $Q = A_1^{\diamond}A^{\dagger}$.*

*Proof.* Let $Y = A^{\dagger,\diamond}$. Denote $P = A^{\dagger}A_1^{\diamond}$, $Q = A_1^{\diamond}A^{\dagger}$. From [12, Proposition 3.1], $P^2 = P$, $Q^2 = Q^* = Q$, i.e., both $P$ and $Q$ are idempotents. From [12, Theorem 3.1], it follows that $PY = A^{\dagger}A_1^{\diamond}A^{\dagger}A_1^{\diamond}A^{\dagger} = P^2A^{\dagger} = PA^{\dagger} = A^{\dagger}A_1^{\diamond}A^{\dagger} = Y$, and $YQ = A^{\dagger}A_1^{\diamond}A_1^{\dagger}A_1^{\diamond}A_1 = A^{\dagger}Q^2 = A^{\dagger}Q = A^{\dagger}A_1^{\diamond}A^{\dagger} = Q$. That is, $PY = Y$

and $YQ = Y$. From [12, Proposition 3.1 and Theorem 3.2] and (1.1), it follows that

$$YAP = U\begin{bmatrix} K^*K & K^*K(\Sigma K)^\dagger \Sigma L \\ L^*K & L^*K(\Sigma K)^\dagger \Sigma L \end{bmatrix} U^* = A^\dagger A_1^\diamond = P,$$

$$QAY = U\begin{bmatrix} \Sigma K(\Sigma K)^\dagger & 0 \\ 0 & 0 \end{bmatrix} U^* = A_1^\diamond A^\dagger = Q.$$

□

**Corollary 2.2.** *A matrix $Y \in \mathbb{C}^{n\times n}$ is the MPBT inverse of $A$, if and only if, $Y$ satisfies $A^\dagger A_1^\diamond Y = Y, YAA^\diamond = Y, YAA^\dagger A_1^\diamond = A^\dagger A_1^\diamond$, and $AA^\diamond AY = AA^\diamond$.*

*Proof.* It is trivial by Theorem 2.1 and the definition of the Bott-Duffin $(P, Q)$-inverse. □

The fundamental algebraic properties inherent in the MPBT inverse itself are first analyzed in depth. These properties cover not only the projection characteristics exhibited by the generalized inverse under left and right multiplication but also reveal its intrinsic connections with the Moore-Penrose inverse and the BT-inverse. A systematic characterization of these algebraic structures not only helps deepen the understanding of the geometric meaning of this new type of generalized inverse but also lays an indispensable theoretical foundation for subsequently converting it equivalently into a problem of solving linear matrix equations and further establishing the convergence analysis of the GNNSYS model. $AA^{\dagger,\diamond}$ is an orthogonal projection operator, i.e., $A^{\dagger,\diamond}$ has an orthogonal projection structure under right multiplication. Property (4) in [12, Proposition 3.1] shows that $A^{\dagger,\diamond}A$ is an idempotent matrix, i.e., $A^{\dagger,\diamond}$ has a projection structure under left multiplication. These two properties respectively characterize the projection features of the MPBT inverse in the left and right directions, and serve as key foundations for the subsequent convergence analysis of GNNSYS.

The matrix $A_1^\diamond$ serves as a key intermediate quantity in the study of the MPBT inverse. Since the importance of outer inverses for generalized inverses was shown in [3], we discuss the outer invertibility of $A_1^\diamond$ in the following proposition.

**Proposition 2.3.** *Let $A \in \mathbb{C}^{n\times n}$. Then $A_1^\diamond \in A^\dagger\{2, 4\}$, i.e., $A_1^\diamond$ is a $\{2, 4\}$-inverse of $A^\dagger$, satisfying*

$$A_1^\diamond A^\dagger A_1^\diamond = A_1^\diamond, (A^\dagger A_1^\diamond)^* = A^\dagger A_1^\diamond.$$

On the basis of the aforementioned fundamental algebraic properties, a further equivalent characterization of the MPBT inverse is given from the broader algebraic framework of inverses along two matrices, laying the foundation for subsequently transforming it into a problem of solving linear matrix equations. The gradient neural network (GNNSYS) model is introduced next to provide an efficient, parallel, and self-stabilizing numerical solution path for the MPBT inverse. To construct this computational model, the primary task is to establish the correspondence between the MPBT inverse and specific matrix equations. If the system $AX = A_1^\diamond A^\dagger$ and $XA = A^\dagger A_1^\diamond$ is consistent, its solution set is an affine subspace; if the solution is unique, then this solution is exactly the MPBT inverse $A^{\dagger,\diamond}$. Before applying the GNNSYS model to system $AX = A_1^\diamond A^\dagger$ and $XA = A^\dagger A_1^\diamond$, the three compatibility conditions need to be verified according to the relevant theorem.

**Theorem 2.4.** *The system of equations $AX = A_1^\diamond A^\dagger$ and $XA = A^\dagger A_1^\diamond$ satisfies the Mitra compatibility conditions:*

$$AA^\dagger A_1^\diamond A^\dagger = A_1^\diamond A^\dagger, A^\dagger A_1^\diamond A^\dagger A = A^\dagger A_1^\diamond, AA^\dagger A_1^\diamond = A_1^\diamond A^\dagger A.$$

*Proof.* It can be directly calculated using the relational expression $AA^{\diamond} = A_1^{\diamond}A^{\dagger}$ and $A^{\dagger}AA^{\diamond}A = A^{\dagger}A_1^{\diamond}$. □

The system of equations $AX = A_1^{\diamond}A^{\dagger}$ and $XA = A^{\dagger}A_1^{\diamond}$ satisfies the Mitra compatibility conditions, the corresponding GNNSYS dynamical model is constructed as $GNNSYS(A, A_1^{\diamond}A^{\dagger}, A, A^{\dagger}A_1^{\diamond})$. According to the framework of J. Dakić and M.D. Petković [2], the dynamic equation of the GNNSYS model is given by

$$\dot{V}(t) = \gamma(A^T\mathcal{F}(A_1^{\diamond}A^{\dagger} - AV(t)) + \mathcal{F}(A^{\dagger}A_1^{\diamond} - V(t)A)A^T), \tag{2.1}$$

where $\gamma > 0$ is a scaling parameter, $\mathcal{F}(\cdot)$ is an element-wise activation function, and $V(t) \in \mathbb{R}^{n\times n}$ is the state matrix.

**Theorem 2.5.** *Let $A \in \mathbb{C}^{n\times n}$, the activation function $\mathcal{F}(\cdot)$, be a monotonically increasing odd function, and $\gamma > 0$ be an arbitrary real number. Taking the initial value $V(0) = 0$, the state matrix $V(t)$ defined by the GNNSYS model* (2.1) *converges to the MPBT inverse as $t \to +\infty$, i.e.,*

$$\lim_{t\to+\infty} V(t) = A^{\dagger,\diamond}.$$

*Proof.* By Theorem 2.4, the system of $AX = A_1^{\diamond}A^{\dagger}$ and $XA = A^{\dagger}A_1^{\diamond}$ satisfies the Mitra compatibility conditions. By [2, Theorem 2.2], when $V(0) = 0$, the limit value is

$$\tilde{V}_0 = A^{\dagger}A_1^{\diamond}A^{\dagger} + A^{\dagger}A_1^{\diamond}A^{\dagger} - A^{\dagger}A_1^{\diamond}A^{\dagger}AA^{\dagger}.$$

Termwise calculation:

$$A^{\dagger}A_1^{\diamond}A^{\dagger} = A^{\dagger}A_1^{\diamond}A^{\dagger}, A^{\dagger}A_1^{\diamond}A^{\dagger} = A^{\dagger}A_1^{\diamond}A^{\dagger}, A^{\dagger}A_1^{\diamond}A^{\dagger}AA^{\dagger} = A^{\dagger}A_1^{\diamond}A^{\dagger}.$$

Therefore

$$\tilde{V}_0 = A^{\dagger}A_1^{\diamond}A^{\dagger} + A^{\dagger}A_1^{\diamond}A^{\dagger} - A^{\dagger}A_1^{\diamond}A^{\dagger} = A^{\dagger}A_1^{\diamond}A^{\dagger} = A^{\dagger,\diamond},$$

hence $\lim_{t\to+\infty} V(t) = A^{\dagger,\diamond}$. □

Up to this point, through the logic of characterizing the MPBT inverse as the solution of a linear matrix equation system, verifying the Mitra compatibility condition, constructing the GNNSYS dynamical equation, and proving convergence under the zero initial condition, this section fully establishes a continuous-time dynamical system computational framework for the MPBT inverse. This framework, guaranteed by Lyapunov stability, gives

$$A^{\diamond} = U\begin{bmatrix} (\Sigma K)^{\dagger} & 0 \\ 0 & 0 \end{bmatrix}U^*$$

whose structure is completely determined by $U$ and $\Sigma K$, independent of $L$. As long as the first $k$ principal singular values of $A$ and the corresponding singular vectors (i.e., the first $k$ columns of $U$, $V$, $\Sigma$) can be efficiently obtained, the BT-inverse can be reconstructed.

**Theorem 2.6.** *Let $A \in \mathbb{C}^{n\times n}$ with $A \approx U\Sigma V^*$ by approximately decomposed by RSVD, where $U, V \in \mathbb{C}^{n\times k}$ are unitary, and $\Sigma \in \mathbb{R}^{k\times k}$ is a diagonal singular value matrix. Let $K = V^*U$. Then the RSVD approximation of the BT-inverse is*

$$A^{\diamond} \approx U(\Sigma K)^{\dagger}U^*.$$

*Proof.* From $U^*U = E_k$, it follows that the column space of $U$ satisfies $\mathcal{R}(A) \subseteq \mathcal{R}(U)$. Denote $\tilde{A} = U\Sigma V^*$, then $\tilde{A}^\dagger = V\Sigma^{-1}U^*, P_{\tilde{A}} = \tilde{A}\tilde{A}^\dagger = UU^*$. Hence $\tilde{A}P_{\tilde{A}} = U\Sigma V^*UU^* = U(\Sigma V^*U)U^* = U(\Sigma K)U^*$. Since $U^*U = E_k$, it holds that $(UMU^*)^\dagger = UM^\dagger U^*$.Therefore,

$$A^\diamond \approx (\tilde{A}P_{\tilde{A}})^\dagger = U(\Sigma K)^\dagger U^*$$

by the definition of Moore-Penrose inverse of $\tilde{A}P_{\tilde{A}}$ and $U^*U = E_k$. □

Based on Theorem 2.6, the RSVD algorithm procedure for the BT-inverse, inspired by [7], is presented as Algorithm 2.1

| | |
|---|---|
| **Input:** | square matrix $A \in \mathbb{C}^{n\times n}$; target rank $k \ll n$; oversampling parameter $p$ |
| **Output:** | approximate matrix $X \approx A^\diamond$ of the BT-inverse |
| **Step 1:** | Generate a Gaussian random projection matrix $\Omega \in \mathbb{C}^{n\times(k+p)}$ |
| **Step 2:** | Compute the sample matrix $Y = A\Omega$ |
| **Step 3:** | $Y = QR$ to obtain the orthogonal matrix $Q \in \mathbb{C}^{n\times(k+p)}$ |
| **Step 4:** | Using $B = Q^*A \in \mathbb{C}^{(k+p)\times n}$ calculate SVD of $B$: $B = \hat{U}\Sigma_{\text{full}}V^*_{\text{full}}$ |
| **Step 5:** | First $k$ principal components, taking $U = (Q\hat{U})_{:,1:k}$, $V = (V_{\text{full}})_{:,1:k}$, $\Sigma = (\Sigma_{\text{full}})_{1:k,1:k}$ |
| **Step 6:** | Compute $K = V^*U$ and output $X = U(\Sigma K)^\dagger U^*$ |

Step 6 in Algorithm 2.1 is a step that is easily overlooked in theoretical analysis but is crucial in implementation. The oversampling parameter $p$ in RSVD is used to improve the approximation accuracy of the principal singular values, but these additional $p$ directions correspond to the null space or near-null space of $A$. Under exact computation, the singular values corresponding to these $p$ directions are exactly zero; under floating-point arithmetic, they are small quantities of the order $10^{-15}$ to $10^{-17}$. In this case, the diagonal matrix $\Sigma$ contains about $p$ nearly zero diagonal entries, and its condition number is as high as $10^{16}$ or even larger. Consequently, the matrix product $\Sigma K$ is severely ill-conditioned, and the numerical computation of its Moore-Penrose inverse $(\Sigma K)^\dagger$ dramatically amplifies noise. The final output $X$ can have a relative error close to $10^0$, leading to complete failure.

After truncation, only $k$ nonzero principal singular values are retained in $\Sigma$. The condition number of $\Sigma K$ drops to the order of $10^2$, and the numerical computation of the Moore-Penrose inverse becomes stable, with a relative error reaching machine precision $10^{-14}$. Two possible approaches exist for the RSVD implementation of the MPBT inverse. The first approach is based on [12, Theorem 3.2]. The second approach uses the simplified form $A^{\dagger,\diamond} = A^\dagger AA^\diamond$, constructing approximations of $A^\dagger$ and $A^\diamond$ from the RSVD factors and then performing matrix multiplication. The first approach is more compact in the Hartwig-Spindelböck structure, but the lower-left block $L^*K$ requires explicit knowledge of $L$, which cannot be directly obtained in the RSVD framework, as RSVD returns only the principal components of $A$ without distinguishing between $U$ and $V$ in the Hartwig-Spindelböck decomposition. The second approach directly utilizes the RSVD outputs $U, V, \Sigma$ to construct approximations of $A^\dagger$ and $A^\diamond$, making the implementation more straightforward. The second approach is adopted in this paper.

**Theorem 2.7.** *Let $A$ be approximately decomposed by RSVD as $A \approx U\Sigma V^*$. Then the RSVD approximation of the MPBT inverse is*

$$A^{\dagger,\diamond} \approx \underbrace{V\Sigma^{-1}U^*}_{\approx A^\dagger} \cdot A \cdot \underbrace{U(\Sigma K)^\dagger U^*}_{\approx A^\diamond},\ K = V^*U.$$

Theorem 2.7 is of considerable practical and theoretical significance. It derives an explicit randomized singular value decomposition approximation for the MPBT inverse, bridging randomized low-rank approximation techniques and the theory of MPBT generalized inverses. The formula clearly demonstrates that the approximation of the MPBT inverse can be assembled from RSVD approximations of $A^{\dagger}$ and $A^{\diamond}$. For large matrices, direct computation of the MPBT inverse is usually expensive; this theorem supplies a low-rank alternative to reduce computational complexity. Moreover, it opens new avenues to study approximation error bounds, parameter selection for RSVD, and fast iterative algorithms for MPBT inverses in high-dimensional applications.

Based on Theorem 2.7, Algorithm 2.2 is obtained.

| | |
|---|---|
| **Input:** | square matrix $A \in \mathbb{C}^{n\times n}$; target rank $k \ll n$; oversampling parameter $p$ |
| **Output:** | approximate matrix $X \approx A^{\dagger,\diamond}$ of the MPBT inverse |
| **Step 1-6:** | Same as Steps 1-6 in Algorithm 2.1, obtaining truncated $U, V \in \mathbb{C}^{n\times k}$, $\Sigma \in \mathbb{R}^{k\times k}$ |
| **Step 7:** | Construct the Moore-Penrose inverse approximation $A^{\dagger} = V\Sigma^{-1}U^{*}$ |
| **Step 8:** | Compute $K = V^{*}U$ and construct the BT-inverse approximation $A^{\diamond} = U(\Sigma K)^{\dagger}U^{*}$ |
| **Step 9:** | Output $X = A^{\dagger}AA^{\diamond}$ |

Algorithm 2.2 shares the first six steps of RSVD with Algorithm 2.1, differing only in the final construction of the generalized inverse. This structural sharing allows the RSVD procedure to be run only once in scenarios where both the BT-inverse and the MPBT inverse are required, followed by the parallel construction of the two generalized inverses, thereby further improving computational efficiency.

## 3. Randomized algorithms of the MPBT-inverse

This section conducts systematic numerical simulation experiments to comprehensively validate and evaluate the theoretical frameworks and algorithmic schemes established in the previous two chapters. Regarding the experimental environment and parameter settings, all experiments are carried out on the MATLAB 2022b platform, and the ordinary differential equation systems are solved using the ode23s stiff solver to ensure the accuracy and stability of the numerical integration of the GNN dynamical equations. According to Section 1, three activation functions are uniformly adopted. This construction exhibits better numerical stability in high dimensions compared to the unitary orthogonal form $U_r\Sigma V_r^{T}$. For examples with special structures, such as the all-ones matrix, the matrix definition is given directly. Throughout this chapter, the relative error in the Frobenius norm is adopted as the accuracy metric:

$$Err(X, X_{ref}) = \frac{\| X - X_{ref} \|_F}{\| X_{ref} \|_F}.$$

For the GNN convergence process, the absolute error $\| V(t) - A^{\dagger,\diamond} \|_F$ is used.

This section examines the reliability of the numerical algorithms for the BT-inverse and MPBT inverse from two independent computational perspectives. First, the theoretical result that the GNNSYS model converges to the MPBT inverse under zero initial condition is verified through typical numerical examples. Then, the high-precision convergent solution of the GNN is used as an independent reference benchmark to systematically cross-validate the output of the RSVD randomized algorithm.

Let the MPBT inverse of the same matrix $A$ be computed by three methods: the direct solution $X_{Direct}$, the GNN convergent solution $X_{GNN}$, and the RSVD approximate solution $X_{RSVD}$. The core proposition of cross-validation is: if the errors of the three pairs $(X_{Direct}, X_{GNN})$, $(X_{GNN}, X_{RSVD})$, and $(X_{Direct}, X_{RSVD})$ are simultaneously of the order $10^{-15}$, then the reliability of the RSVD algorithm can be mutually corroborated from three independent perspectives.

This paper uses the GNN method established in Section 2 as an independent high-precision reference benchmark to construct a GNN-RSVD cross-validation framework. First, the convergence verification of the GNNSYS model is presented.

(1) Experimental setup: Consider a matrix $A \in \mathbb{C}^{3\times 3}$.

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad rank(A) = 1.$$

Direct computation yields its MPBT inverse as $A^{\dagger} = \frac{1}{9}A$, and the parameter matrices $C_0 = AA^{\circ}AA^{\dagger}$ and $D_0 = A^{\dagger}AA^{\circ}A$ can be computed from the known matrix $A$.

(2) Numerical verification of the Mitra compatibility condition: By Theorem 2.4, the condition for the GNNSYS model to be compatible in this example is that the three Mitra conditions hold. Numerical computation shows that all three residuals are of the order $10^{-15}$, strictly satisfying the Mitra compatibility condition, thus providing numerical guarantee for the GNNSYS model convergence.

(3) Convergence process under three activation functions: With scaling parameter $\gamma = 100$ with $V(0) = 0$, the GNNSYS model is run using three activation functions and the evolution curves of the error norm over time are shown in Figure 1.

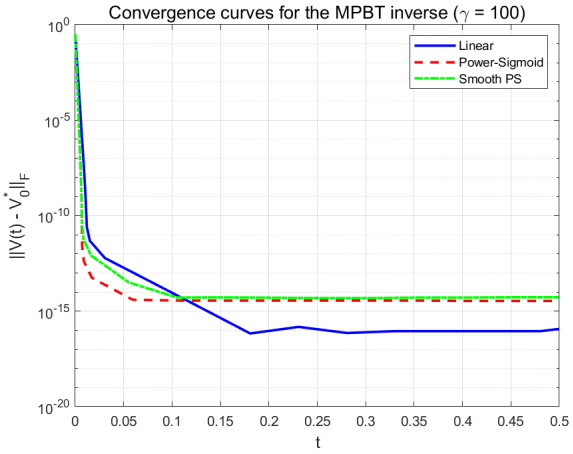



**Figure 1.** Evolution curves of the MPBT inverse under three activation functions.

Parameter tests show that $\gamma = 100$ achieves the best comprehensive performance of convergence speed and steady-state error. A larger $\gamma$ accelerates convergence, yet an excessively large value may cause oscillation. Thus, $\gamma = 100$ is selected as a trade-off.

Under the three activation functions, the model error decreases smoothly from the initial value and eventually stabilizes at the order of machine precision, numerically confirming the correctness of Theorem 2.5.

(1) Experimental setup: A random low-rank matrix with $n = 10$, $r = 6$ is taken, and the relative errors of the three methods are computed and shown in the following table.

As shown in Table 1, the results of the three methods are highly consistent with each other, and the relative errors are all near machine precision. The relative error between GNN and Direct reaches $10^{-14}$, while the relative errors between RSVD and Direct and between RSVD and GNN are both of the order $10^{-13}$. The correctness of the RSVD algorithm is mutually corroborated from two independent computational paradigms.

**Table 1.** Cross comparison of methods

| Method | Time(s) | Direct comparison | GNN comparison |
|---|---|---|---|
| GNN | 195.28 | $1.34 \times 10^{-14}$ | – |
| RSVD | 0.0018 | $1.70 \times 10^{-13}$ | $1.69 \times 10^{-13}$ |

(2) Evolution of the gap between the GNN trajectory and the RSVD result: To further demonstrate the consistency between the GNN and RSVD methods throughout the entire computational process, the evolution curves of the relative distances between the GNN state matrix $V(t)$ during integration and the direct reference solution as well as the RSVD result over time are plotted, as shown in Figure 2.

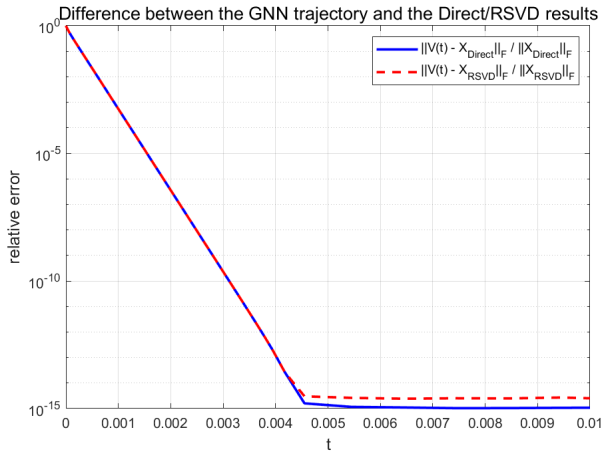



**Figure 2.** Evolution of the gap between the GNN trajectory and the direct/RSVD results

The two curves essentially coincide, indicating that the distance between the GNN and the RSVD result throughout the entire evolution path is always approximately equal to the distance from the GNN itself to the true solution. This shows that, relative to the GNN trajectory, the RSVD result can be regarded as a point within the machine-precision neighborhood of the GNN final convergence point, further corroborating the equivalent reliability of the two methods. The convergence of the GNNSYS model itself has been rigorously verified under multiple activation functions; using the GNN convergent solution as a reference benchmark, the output of the RSVD algorithm is consistent with it at the order of machine precision, confirming the reliability of the randomized algorithm from two mutually independent computational paradigms.

The approximation error of the RSVD algorithm mainly originates from the degree to which the random projection captures the low-dimensional subspace of $A$. According to the classic result of

Halko-Martinsson-Tropp [4], the RSVD reconstruction $\tilde{A} = \tilde{U}\Sigma V^*$ satisfies

$$\mathbb{E} \parallel A - \tilde{A} \parallel_F \leqslant \left(1 + \frac{k}{p-1}\right)^{\frac{1}{2}} \left(\sum_{j=k+1}^{n} \sigma_j^2\right)^{\frac{1}{2}}, \ p \geqslant 2.$$

When $A$ has low rank, $\sum_{j=k+1}^{n} \sigma_j^2$ is very small, and the reconstruction error of RSVD is correspondingly small. Generalized inverses are sensitive to perturbations. The perturbation of the BT-inverse satisfies

$$\parallel \tilde{A}^{\diamond} - A^{\diamond} \parallel_F \leqslant C_2 \kappa(AP_A)\varepsilon.$$

For the MPBT inverse, since its expression $A^{\dagger,\diamond} = A^{\dagger}AA^{\diamond}$ involves the product of two generalized inverses, the perturbation error is approximately the sum of two terms:

$$\parallel \tilde{A}^{\dagger,\diamond} - A^{\dagger,\diamond} \parallel_F \lesssim \parallel \tilde{A}^{\dagger} - A^{\dagger} \parallel_F \parallel AA^{\diamond} \parallel_F + \parallel A^{\dagger}A \parallel_F \parallel \tilde{A}^{\diamond} - A^{\diamond} \parallel_F .$$

The symbol $\lesssim$ denotes the dominant upper bound of the error after neglecting higher-order small quantities. On the basis of confirming the reliability of the algorithm, this section further systematically investigates the key factors affecting numerical performance and their error distribution patterns.

(1) Experimental setup: Take a $4 \times 4$ random matrix. The true value $A^{\dagger,\diamond}$ of the MPBT inverse is obtained by the direct method as a reference.

(2) Influence of $\gamma$ on convergence speed: Using the linear activation function, take $\gamma = 10, 100, 1000$ respectively, with initial value $V(0) = 0$. Plot the evolution curves of the error norm $\parallel V(t) - A^{\dagger,\diamond} \parallel_F$.

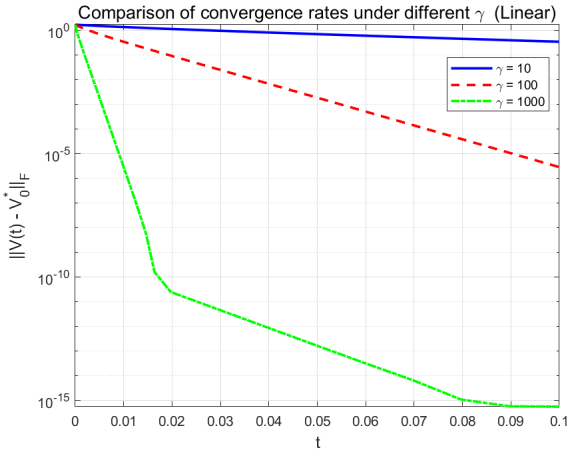



**Figure 3.** Comparison of convergence speeds of the GNNSYS model

Since $\gamma$ acts as a time scaling factor in the GNNSYS dynamical equation $\dot{V} = \gamma\mathcal{F}(\cdot)$, a large $\gamma$ is equivalent to integrating the original system to a much later time, so the convergence time is inversely proportional to $\gamma$.

(3) Element-wise convergence trajectory: Taking $\gamma = 100$, the evolution curves of the 16 components $v_{ij}(t)$ of the state matrix $V(t)$ over time are plotted.

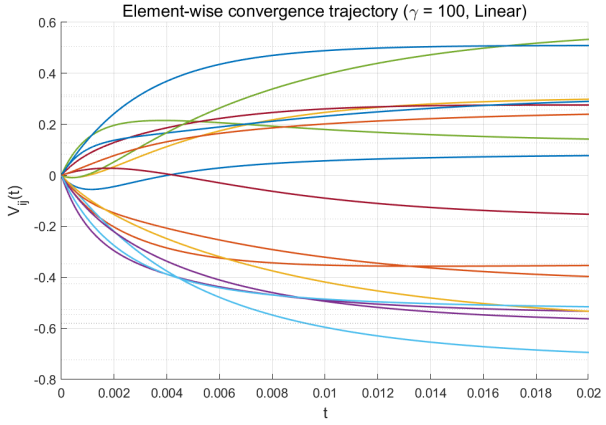



**Figure 4.** Element-wise convergence trajectories of the MPBT inverse

All elements start from zero initial values, gradually differentiate under the driving of the dynamical system, and eventually converge smoothly to the steady-state values at the corresponding positions in the true value $A^{\dagger,\diamond}$. There exist significant differences in the convergence rates among the elements: some elements increase monotonically, while others decrease monotonically. This diversity is related to the spectral structure of the matrix $A$. No obvious oscillations are observed under linear activation, indicating that the system has good stability. This experiment verifies the correctness and efficiency advantage of Algorithm 2.1 in Section 2, and systematically compares it with the direct method $A^{\diamond} = (AP_A)^{\dagger}$.

(1) Small-scale correctness verification: Take a random low-rank matrix with $n = 50$, $r = 30$, and $p = 10$. After running Algorithm 2.1 and the direct method, the relative error reaches $10^{-14}$, which is within the accuracy range. This verifies the correctness of Algorithm 2.1 based on the formula $A^{\diamond} = U(\Sigma K)^{\dagger}U^{*}$.

(2) Efficiency and accuracy comparison at different scales: Take matrix rank $r = 0.6n$ and test for $n = 100, 200, 400, 600, 800$. The running times of the two methods and the relative error of RSVD are shown in Table 2.

**Table 2.** Comparison of direct and RSVD methods for the BT-inverse

| $n$ | $r$ | Direct time (s) | RSVD time (s) | Relative error |
|---|---|---|---|---|
| 100 | 60 | 0.0060 | 0.0048 | $1.7659 \times 10^{-12}$ |
| 200 | 120 | 0.0166 | 0.0132 | $4.2400 \times 10^{-12}$ |
| 400 | 240 | 0.0667 | 0.0400 | $3.1978 \times 10^{-13}$ |
| 600 | 360 | 0.1533 | 0.0997 | $2.3434 \times 10^{-13}$ |
| 800 | 480 | 0.3224 | 0.1871 | $5.8627 \times 10^{-13}$ |

To comprehensively evaluate the computational efficiency and approximation accuracy of the MPBT and BT algorithms in matrix inversion problems, runtime comparison plots (Figure 5) and RSVD relative error comparison plots (Figure 6) are respectively generated, which collectively illustrate the performance differences of the two algorithms across various matrix dimensions $n$. Through the integrated comparison of these two figures, the trade-off between direct methods and RSVD approximations in terms of time efficiency and accuracy can be intuitively analyzed, along with the applicability of the MPBT-inverse and BT-inverse algorithms to problems of different scales.

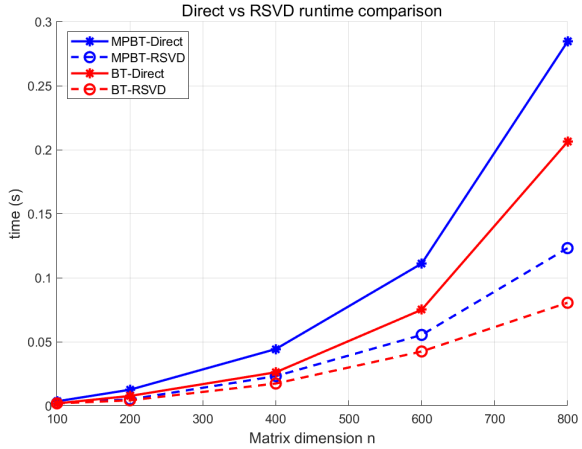



**Figure 5.** Comparison of running time between the direct method and RSVD for the MPBT-inverse.

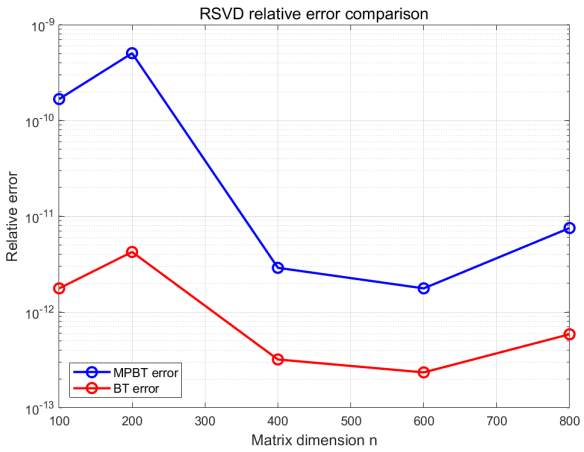



**Figure 6.** Variation of relative error of RSVD for the MPBT-inverse with matrix dimension.

Based on the comparative results from Figures 5 and 6, the following analysis can be drawn:

(1) In terms of runtime (Figure 5), for both MPBT and BT algorithms, the execution time of the direct method exhibits an approximately quadratic or even cubic growth trend as the matrix dimension $n$ increases. In contrast, the runtime of the RSVD approximation method is significantly smoother, with a much lower growth rate, which fully demonstrates the computational advantage of RSVD in low-rank approximation solutions. When comparing the two algorithms, MPBT and BT show similar time consumption under the direct method; however, under the RSVD mode, BT-RSVD is slightly faster than MPBT-RSVD, suggesting that the BT algorithm may possess a lighter computational structure when combined with RSVD. Furthermore, the time gap between the direct method and RSVD widens dramatically with increasing $n$, indicating that RSVD approximation can substantially enhance solving efficiency for large-scale problems.

(2) In terms of approximation accuracy (Figure 6), the relative errors of RSVD for both MPBT and BT remain at extremely low levels (approximately $10^{-12}$ order of magnitude) and stay essentially stable as the dimension grows, without noticeable error accumulation. This indicates that both algorithms can provide high-fidelity estimates of the inverse matrix under RSVD approximation. Among them, MPBT

exhibits slightly lower error than BT, but the difference remains within the same order of magnitude, so their approximation accuracy can be considered comparable in practical applications.

In summary, for matrix inversion problems, the RSVD approximation method can substantially reduce computational cost with almost no sacrifice in accuracy, making it particularly suitable for large-scale, low-rank structured problems. The direct method performance of MPBT and BT is similar; however, with RSVD acceleration, BT requires less time, while MPBT achieves higher accuracy. The overall results indicate that the RSVD-based approximate inversion strategy is an effective approach that combines high efficiency and reliability.

## 4. Conclusions

The present results can stimulate several further developments. First, the proof strategy for the link between MPBT invertible matrices and Bott-Duffin invertibility may be extended to other types of generalized inverses. Second, the gradient neural network framework developed here can be adapted to solve more complicated coupled matrix equations involving generalized inverses. Third, the GNNSYS model provides a candidate dynamic solver for related practical problems such as constrained optimization and singular system control. There remain several open questions to be addressed in subsequent work. The convergence of the proposed GNNSYS model is analyzed under Mitra compatibility conditions; it is still unclear how to modify the network structure for incompatible matrix systems. In addition, convergence speed, robustness against noise and external disturbances of the GNNSYS model require more rigorous theoretical analysis and experimental validation. Moreover, the properties of MPBT inverses for non-square matrices and the connections between MPBT invertibility and other hybrid Bott-Duffin type invertibility concepts are yet to be fully explored.

The main tools of our article are matrices and operations on them, in particular matrix multiplication. There are a variety of fast algorithms to this problem. For example, a modern algorithm, which importantly is practical in implementations, is given by the recent work in [9].

## Author contributions

Sanzhang Xu: Writing-original draft; Zhengyang Shan and Can Zha: Writing-review and editing. All authors have read and approved the final version of the manuscript for publication.

## Use of Generative-AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Acknowledgments

The research article is supported the Qing Lan Project of Jiangsu Province and the Natural Science Foundation of Huai'an of Jiangsu Province (No.HAB2025003).

## Conflict of interest

The authors declare that there is no conflict of interest.

## References


1. O. M. Baksalary, G. Trenkler, On a generalized core inverse, *Appl. Math. Comput.*, **236** (2014), 450–457. http://doi.org/10.1016/j.amc.2014.03.048
2. J. Dakić, M. D. Petković, Gradient neural network model for the system of two linear matrix equations and applications, *Appl. Math. Comput.*, **481** (2024), 128930. https://doi.org/10.1016/j.amc.2024.128930
3. M. P. Drazin, A class of outer generalized inverses, *Linear Algebra Appl.*, **436** (2012), 1909–1923. https://doi.org/10.1016/j.laa.2011.09.004
4. N. Halko, P. G. Martinsson, J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Rev.*, **53** (2011), 217–288. https://doi.org/10.1137/090771806
5. R. A. Horn, C. R. Johnson, *Matrix Analysis*, Cambridge: Cambridge University Press, 2012.
6. R. E. Hartwig, K. Spindelböck, Matrices for which $A^*$ and $A^\dagger$ commmute, *Linear Multilinear A.*, **14** (1983), 241–256. https://doi.org/10.1080/03081088308817561
7. H. F. Ma, H. B. Chen, P. S. Stanimirović, Representations and computation of the BT-inverse and its randomized algorithms, *Appl. Math. Comput.*, **515** (2026), 129850. https://doi.org/10.1016/j.amc.2025.129850
8. R. Penrose, A generalized inverse for matrices, *Proc. Cambridge Philos. Soc.*, **51** (1955), 406–413. https://doi.org/10.1017/S0305004100030401
9. J. S. Respondek, Correction of 'J. Laderman, V. Pan, X. H. Sha, On practical algorithms for accelerated matrix multiplication, Linear Algebra and its Applications, Vol. 162-164 (1992), 557–588', *Linear Multilinear Algebra*, **73** (2025), 838–848. https://doi.org/10.1080/03081087.2024.2391807
10. C. R. Rao, S. K. Mitra, Generalized inverse of a matrix and its applications, In: *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*, California: University of California Press, **6** (1972), 601–620. https://doi.org/10.1525/9780520325883-032
11. I. Stojanović, P. S. Stanimirović, I. S. Živković, D. Gerontitis, X. Z. Wang, ZNN models for computing matrix inverse based on hyperpower iterative methods, *Filomat*, **31** (2017), 2999–3014. https://doi.org/10.2298/fil1710999s
12. S. Z. Xu, Q. Y. Xu, N. Zhou, The MPBT inverse of a complex matrix based on the Hartwig-Spindelböck decomposition, *Ital. J. Pure Appl. Math.*, **52** (2024), 205–218.


AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)