

ALGORITHMIC COMPLEXITY ANALYSIS AND NUMERICAL EXPERIMENTS FOR COMPUTING THE MATRIX EXPONENTIAL

Hang Ye*, Hua Guo, ZeDong Li
Aviation University of Air Force, Changchun 130000, Jilin, China.
*Corresponding Author: Hang Ye

Abstract: The fundamental solution matrix is a central object in the theoretical study and numerical computation of linear systems of ordinary differential equations with constant coefficients. This paper establishes a unified framework for the algorithmic complexity analysis of seven classical methods, with theoretical comparison from the perspectives of time complexity, space complexity, and numerical stability. Numerical experiments on six classes of typical test matrices are conducted, and quantitative evaluations are performed with respect to computational accuracy, running time, and robustness. The results indicate that the scaling-and-squaring method achieves optimal numerical stability; the eigenvalue decomposition method attains the highest accuracy for diagonalizable well-conditioned matrices; and the direct sum decomposition method significantly outperforms the Jordan canonical form method on non-diagonalizable matrices. A selection strategy is proposed to provide reference for teaching and engineering practice.

Keywords: Fundamental solution matrix; Matrix exponential function; Algorithmic complexity; Numerical stability; Numerical experiment

1 INTRODUCTION

The computation of the fundamental solution matrix is a core topic in ordinary differential equations and a key computational component in control theory, quantum mechanics, circuit analysis, and related fields [1]. Shang and Zhang discussed representations of exponential matrices from the perspective of differential-equation solutions [2], revealing the close connection between fundamental solution matrices and differential-equation theory. Different computational methods differ substantially in theoretical elegance, computational efficiency, and numerical stability. Existing textbooks and literature usually emphasize the derivation of a single method and lack systematic algorithmic-complexity analysis and cross-method numerical comparisons, making method selection difficult in teaching and practice.

In recent years, substantial progress has been made in computing matrix exponentials. Blanes et al. proposed a new scaling-and-squaring method in 2025 that combines Taylor expansions with optimized Taylor-polynomial approximations and outperforms existing software implementations [3]. Pozza and Van Buggenhout developed a Legendre-polynomial-based method that provides an efficient approach to computing exponentials of large sparse matrices [4]. Yao et al. systematically reviewed extensions of the precise integration method to time-varying systems [5], nonlinear problems, and large-scale computation, demonstrating its continuing potential. These advances show that matrix-exponential computation remains active; however, the classical methods have not yet been compared systematically from the perspective of teaching and engineering practice.

To address these issues, this paper establishes a unified complexity-analysis framework for seven methods of computing fundamental solution matrices. A test-matrix set covering extreme cases such as ill-conditioned and non-diagonalizable matrices is designed. Reproducible numerical experiments reveal the accuracy limits of the methods, and a method-selection strategy is proposed on the basis of the experimental results for teaching and engineering practice.

2 FUNDAMENTAL SOLUTION MATRIX METHODS AND COMPLEXITY ANALYSIS

The matrix exponential is defined by

$$e^{tA} = \sum_{k=0}^{\infty} \frac{(tA)^k}{k!} \quad (1)$$

This power series converges uniformly for every finite-dimensional matrix A and $t \in \mathbb{R}$. The following methods are developed from this definition or equivalent forms.

2.1 Direct Expansion Methods

Power-series expansion. Truncating Equation (1) directly gives

$$e^{tA} \approx \sum_{k=0}^K \frac{(tA)^k}{k!} \quad (2)$$

The truncation error is controlled by $\|tA\|^{K+1} / (K+1)!$. The time complexity is $O(Kn^3)$, and the space complexity is $O(n^2)$. Stability is moderate, but convergence is slow for large-norm matrices.

Hamilton-Cayley method. By the Hamilton-Cayley theorem, e^{tA} can be represented by an A -degree matrix polynomial in $n-1$. The coefficients are determined from the characteristic polynomial by Sylvester interpolation or the undetermined-coefficients method. Repeated eigenvalues require derivative conditions, producing an ill-conditioned linear system. The time complexity is $O(n^4)$ and the space complexity is $O(n^2)$. Stability is poor when eigenvalues are close, and the method can fail for ill-conditioned matrices.

2.2 Matrix-Decomposition Methods

Eigenvalue-decomposition method. When $A = PDP^{-1}$ is diagonalizable, $e^{tA} = Pe^{tD}P^{-1}$. The time complexity is $O(n^3)$, the space complexity is $O(n^2)$. Stability depends on P , namely the condition number $\kappa(P)$; ill conditioning amplifies round-off errors [6].

Jordan canonical-form method. $A = PJP^{-1}$, where J is a direct sum of Jordan blocks. For the i -th order Jordan block m_i ($J_i = \lambda_i I + N_i$ is nilpotent and $N_i^{m_i} = 0$), with $N_i^{m_i} = 0$, the matrix exponential has an explicit upper-triangular Toeplitz structure:

$$e^{tJ_i} = e^{\lambda_i t} \sum_{k=0}^{m_i-1} \frac{t^k}{k!} N_i^k \quad (3)$$

The time complexity is $O(n^3)$. Stability is extremely sensitive to perturbations and is numerically unstable [7].

Direct-sum decomposition method. Let A have distinct eigenvalues $\lambda_1, \dots, \lambda_r$ with corresponding generalized eigenspaces V_i . By direct-sum decomposition, any initial value can be uniquely written as $\mathbb{C}^n = \bigoplus_{i=1}^r V_i$. After the projection operators yield x_0 , each component is computed from a finite exponential polynomial $x_0 = \sum_{i=1}^r x_{0,i}$ of the form $x_{0,i}$:

$$e^{tA} x_{0,i} = e^{\lambda_i t} \sum_{k=0}^{m_i-1} \frac{t^k}{k!} (A - \lambda_i I)^k x_{0,i} \quad (4)$$

Finally, time complexity is $O(n^3)$, the space complexity is $O(n^2)$, and stability is better than that of the Jordan canonical-form method because explicit Jordan decomposition is avoided.

2.3 Numerical Approximation Methods

Scaling-and-squaring method. Choose an integer s such that $\left\| \frac{A}{2^s} \right\|$ is sufficiently small; compute $e^{tA/2^s}$ using a

Padé approximation and then recover the result by s squarings [8-9]. Blanes et al. further improved the method in 2025 by combining Taylor expansions with optimized Taylor-polynomial approximations, improving efficiency while preserving stability [3]. The time complexity is $O(n^3)$, the space complexity is $O(n^2)$, and stability is optimal because the intermediate norm is controlled.

Laplace-transform method. $e^{tA} = \mathcal{L}^{-1}\{(sI - A)^{-1}\}$; numerical inversion is ill-posed. The time complexity is $O(Mn^3)$, where M is the number of samples used in numerical inversion. Stability is very poor, so this method is excluded from the experiments.

2.4 Summary of Complexity and Stability

Table 1 provides the theoretical comparison of the seven methods.

Table 1 Summary of Method Complexity and Stability

Method	Time Complexity	Space Complexity	Stability	Applicable Scenario
Eigenvalue Decomposition	$O(n^3)$	$O(n^2)$	Moderate	Well-conditioned diagonalizable matrices
Jordan Canonical Form	$O(n^3)$	$O(n^2)$	Poor	Theoretical derivation
Direct-Sum Decomposition	$O(n^3)$	$O(n^2)$	Moderate	Non-diagonalizable matrices
Laplace Transform	$O(Mn^3)$	$O(Mn^2)$	Poor	Theoretical analysis; excluded from numerical comparison
Power-Series Expansion	$O(Kn^3)$	$O(n^2)$	Moderate	Small-norm matrices
Hamilton-Cayley	$O(n^4)$	$O(n^2)$	Poor	Manual low-dimensional, well-conditioned calculation
Scaling and Squaring	$O(n^3)$	$O(n^2)$	Excellent	General numerical computation

Note: M is the number of samples used for Laplace inversion, K is the truncation order of the power series, and s is the number of scaling steps.

3 NUMERICAL EXPERIMENTS

3.1 Experimental Environment and Reproducibility

Experiments were performed in Python 3.11 using NumPy 1.26 and SciPy 1.11 on an Intel Core i7-12700H computer with 16 GB RAM. Machine precision is $\varepsilon \approx 2.22 \times 10^{-16}$.

3.2 Test Matrices

Six classes of test matrices were designed: (1) diagonalizable matrices: $A = P\Lambda P^{-1}$, where $\text{diag}(J_2(\lambda), \dots, J_2(\lambda))$ is a random invertible matrix; (2) repeated-eigenvalue matrices: $A = PJP^{-1}$; (3) non-diagonalizable matrices: a Jordan block under the similarity transformation P , where $A = PHP^{-1}D$ is random and invertible; (4) ill-conditioned matrices: H , where the first factor is a Hilbert matrix and D is diagonal with entries spanning several orders of magnitude, so that $\text{cond}(A) \approx 10^{12}$; (5) random matrices: $A \in \mathbb{R}^{n \times n}$ with entries drawn from $N(0, 1)$; and (6) symmetric positive-definite matrices: $A = Q^T Q$, where Q is a random matrix with entries from the standard normal distribution.

This section compares six of the seven methods numerically. The Laplace-transform method produced no valid results because numerical inverse transformation is ill-posed and is therefore excluded from the comparison.

3.3 Evaluation Metrics

Using SciPy `expm` (the scaling-and-squaring implementation) as the reference E_{ref} , the relative error is defined as $\|E - E_{\text{ref}}\|_F / \|E_{\text{ref}}\|_F$; the mean CPU time (seconds, averaged over five runs) is also recorded. The set of time parameters is t . For matrices of dimension $\{0.1, 0.5, 1.0, 2.0, 5.0\}$, use $n = 20$; for matrices of dimension $t = 1.0$, use $n = 50$ and set $t = 0.5$. Tables 2 and 3 report the relative errors and mean CPU times under these settings.

3.4 Experimental Results

Tables 2 and 3 provide the accuracy and efficiency comparisons, respectively. The Hamilton-Cayley method fails on ill-conditioned matrices because nearly coincident eigenvalues make the interpolation conditions ill-conditioned and the linear-system solution fails. The Jordan canonical-form method gives completely distorted results on ill-conditioned matrices and is counted as a failure. These failures are denoted by an em dash in the tables.

Table 2 Mean Relative Errors of the Methods

Matrix Type	Eigenvalue Decomposition	Direct-Sum Decomposition	Power-Series Expansion	H-C Theorem	Scaling and Squaring	Jordan Canonical Form
-------------	--------------------------	--------------------------	------------------------	-------------	----------------------	-----------------------

Matrix Type	Eigenvalue Decomposition	Direct-Sum Decomposition	Power-Series Expansion	H-C Theorem	Scaling and Squaring	Jordan Canonical Form
Diagonalizable	1.2×10^{-14}	8.5×10^{-14}	2.1×10^{-14}	3.5×10^{-13}	2.2×10^{-16}	1.3×10^{-14}
Repeated Eigenvalues	5.8×10^{-15}	7.2×10^{-14}	1.8×10^{-13}	2.4×10^{-8}	2.1×10^{-16}	3.6×10^{-7}
Non-diagonalizable	4.2×10^{-6}	6.8×10^{-13}	5.2×10^{-10}	1.8×10^{-3}	3.1×10^{-16}	2.1×10^{-8}
Ill-conditioned	3.5×10^{-6}	2.1×10^{-12}	1.2×10^{-14}	—	3.8×10^{-16}	—
High-dimensional Random	2.8×10^{-15}	5.4×10^{-13}	3.6×10^{-9}	4.2×10^{-2}	2.9×10^{-16}	8.1×10^{-12}
Symmetric Positive-definite	1.5×10^{-15}	3.2×10^{-14}	8.5×10^{-11}	5.7×10^{-9}	2.5×10^{-16}	1.6×10^{-15}

Note: An em dash indicates that the method failed on this class of test matrices because of numerical instability or iterative divergence.

Table 3 Mean Computation Times of the Methods (s)

Matrix Type	Eigenvalue Decomposition	Direct-Sum Decomposition	Power-Series Expansion	H-C Theorem	Scaling and Squaring	Jordan Canonical Form
Diagonalizable	1.2×10^{-4}	2.8×10^{-4}	3.2×10^{-4}	4.5×10^{-4}	3.5×10^{-5}	2.1×10^{-4}
Repeated Eigenvalues	1.1×10^{-4}	2.6×10^{-4}	3.3×10^{-4}	4.2×10^{-4}	3.4×10^{-5}	2.0×10^{-4}
Non-diagonalizable	1.3×10^{-4}	3.1×10^{-4}	3.4×10^{-4}	4.8×10^{-4}	3.6×10^{-5}	2.3×10^{-4}
Ill-conditioned	1.2×10^{-4}	2.9×10^{-4}	3.2×10^{-4}	—	3.7×10^{-5}	—
High-dimensional Random	1.8×10^{-3}	4.2×10^{-3}	6.5×10^{-4}	2.8×10^{-3}	5.8×10^{-5}	1.2×10^{-3}
Symmetric Positive-definite	1.0×10^{-4}	2.4×10^{-4}	3.5×10^{-4}	4.1×10^{-4}	3.4×10^{-5}	1.9×10^{-4}

Note: An em dash indicates that the method failed on this class of test matrices because of numerical instability or iterative divergence.

Main findings: (1) the scaling-and-squaring method achieves machine precision (approximately 10^{-16}) for every test matrix and has the best numerical stability; (2) eigenvalue decomposition is highly accurate on well-conditioned matrices (about 10^{-15}) but its error increases sharply to 10^{-6} on ill-conditioned matrices; (3) direct-sum decomposition maintains an accuracy of 10^{-13} on non-diagonalizable matrices and substantially outperforms the Jordan canonical-form method (10^{-8}); (4) the Hamilton-Cayley method is acceptable for low-dimensional well-conditioned matrices (10^{-13}) but errors increase sharply or complete failure occurs in high-dimensional or ill-conditioned cases; (5) the Jordan canonical-form method produces errors of 10^{-7} for repeated-eigenvalue matrices and fails completely for ill-conditioned matrices; and (6) scaling and squaring has the shortest running time (10^{-5} to 10^{-4} s) and is insensitive to matrix type.

3.5 Effects of Condition Number and Dimension

A condition-number-controlled matrix is constructed as $A = Q^T \text{diag}(1, 10^2, \dots, 10^{2k}) Q$, where Q is orthogonal and $k = 1, 2, 3, 4, 5, 6$. The results show that the error of scaling and squaring remains near machine precision; the error of eigenvalue decomposition grows approximately linearly with condition number; direct-sum decomposition grows more slowly and is superior to eigenvalue decomposition; and the Hamilton-Cayley method fails completely when eigenvalues are highly clustered, with relative error exceeding 10^3 .

Additional symmetric positive-definite matrices $n = 10, 20, 30, 40, 50$ were tested. The results confirm the theoretical time-complexity predictions: the Hamilton-Cayley method grows fastest with dimension, confirming $A = Q^T Q$ complexity; eigenvalue decomposition, direct-sum decomposition, and the Jordan canonical-form method grow approximately as $O(n^4)$; scaling and squaring grows most slowly and remains efficient at $O(n^3)$ with $n = 50$.

4 CONCLUSIONS AND METHOD-SELECTION STRATEGY

The seven classical methods for computing fundamental solution matrices are compared systematically through algorithmic-complexity analysis and numerical experiments. Scaling and squaring performs best in accuracy, efficiency, and stability and is suitable for general numerical computation. Eigenvalue decomposition is highly accurate when the matrix is well-conditioned and diagonalizable, but its error increases sharply in ill-conditioned cases. Direct-sum decomposition is substantially more accurate than the Jordan canonical-form method on non-diagonalizable matrices. Power-series expansion with a fixed truncation order is strongly affected by matrix norm and dimension. The Hamilton-Cayley method has poor stability when eigenvalues are close or the matrix is ill-conditioned and may fail

completely. The Jordan canonical-form method is extremely sensitive to numerical perturbations and has poor stability on ill-conditioned matrices.

Based on the theoretical analysis and numerical experiments, the following method-selection strategy is proposed:

(1) Scaling and squaring is the first choice for general numerical computation. It is optimal in accuracy, efficiency, and stability and is therefore recommended as the default implementation for general scientific-computing tasks.

(2) Eigenvalue decomposition is preferred for theoretical analysis. When the matrix is diagonalizable and $\text{cond}(P) < 10^8$, it is accurate and has a clear physical interpretation. For symmetric positive-definite matrices, orthogonal diagonalization yields the best condition number.

(3) Use direct-sum decomposition for non-diagonalizable matrices. It avoids the numerical instability of the Jordan canonical-form method by determining generalized eigenvectors through linear systems.

(4) The Hamilton-Cayley method may be considered for manual computation of low-dimensional, well-conditioned matrices. For small matrices satisfying $n \leq 4$ and $\text{cond}(P) < 10^4$, direct hand calculation is more convenient than matrix decomposition.

(5) Avoid using the Jordan canonical-form method and the Laplace-transform method alone for numerical computation. Both exhibit serious stability problems and are suitable only for theoretical derivations.

A limitation of this study is that the largest test-matrix dimension is 50. Parallel-computing performance for larger matrices and the special structure of sparse matrices have not been investigated. Future work will investigate Krylov-subspace-based fast algorithms for fundamental solution matrices of large sparse matrices and their integration into teaching-case repositories [10-12].

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

ACKNOWLEDGEMENTS

The authors sincerely thank the relevant literature for its inspiration and the reviewers for their valuable comments.

REFERENCES

- [1] Moler C, Van Loan C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 2003, 45(1): 3-49.
- [2] Shang Y L, Zhang Q L. Representation of the exponential matrix $e^{(A)t}$ for solutions of first-order differential equation systems. *Journal of Luoyang University*, 1995, 10(4): 30-33.
- [3] Blanes S, Kopylov N, Seydaoğlu M. Efficient scaling and squaring method for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 2025, 46(1): 74-93.
- [4] Pozza S, Van Buggenhout N. A new Legendre polynomial-based approach for non-autonomous linear ODEs. *Electronic Transactions on Numerical Analysis*, 2024, 60: 292-326.
- [5] Yao W A, Gao Q, Tan S J, et al. Development and extended applications of the precise integration method. *Chinese Journal of Computational Mechanics*, 2024, 41(1): 1-24.
- [6] Higham N J. *Functions of matrices: theory and computation*. Philadelphia: SIAM, 2008.
- [7] Golub G H, Van Loan C F. *Matrix computations*. 4th ed. Baltimore: Johns Hopkins University Press, 2013.
- [8] Tan S J, Wu Z G, Zhong W X. Adaptive selection of parameters in the precise integration method for matrix exponentials. *Chinese Journal of Theoretical and Applied Mechanics*, 2009, 41(6): 961-966.
- [9] Zhang H W, Zhong W X. Discussion of algorithms for computing matrix exponentials. *Journal of Dalian University of Technology*, 2000, 40(5): 522-525.
- [10] Güttel S, Schweitzer M. Randomized sketching for Krylov approximations of large-scale matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 2023, 44(3): 1073-1095.
- [11] Guidotti N L, Martinsson P G, Acebrón J A, et al. Accelerating a restarted Krylov method for matrix functions with randomization. *arXiv preprint*, arXiv: 2503.22631, 2025.
- [12] Zheng Y. A theorem on a matrix exponential function and its teaching method. *Mathematical Theory and Applications*, 2018, 38(Z2): 123-128.