Mathematics

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

AIMS Mathematics, 10(3): 6546–6559.
DOI: 10.3934/math.2025299
Received: 29 December 2024
Revised: 07 March 2025
Accepted: 17 March 2025
Published: 24 March 2025

Research article

Fast algorithms for a linear system with infinitesimal generator structure of a Markovian queueing model

Jiaqi Qu[1], Yunlan Wei[1,*], Yanpeng Zheng[2,*] and Zhaolin Jiang[1]

[1] School of Mathematics and Statistics, Linyi University, Linyi 276000, China
[2] School of Automation and Electrical Engineering, Linyi University, Linyi 276000, China

* **Correspondence:** Email: wyl19910110@163.com, zhengyanpeng0702@sina.com.

Abstract: In this paper, we focused on solving the perturbed four-banded linear system derived from the traffic process associated with a Markovian queueing model. Utilizing the spectral decomposition of circulant and skew circulant matrices, we computed the product of Toeplitz inversion and a vector, leading to a decomposition algorithm for perturbed four-banded linear systems. This decomposed Toeplitz system features multiple right-hand terms, significantly reducing computational complexity through Toeplitz inversion. Additionally, we introduced an algorithm based on banded LU decomposition, resulting in a banded linear system with multiple right-hand terms, where the sparsity of the banded LU decomposition is pivotal. To evaluate the algorithm's performance, we presented two examples in numerical simulations.

Keywords: band LU decomposition; infinitesimal generator structure; Markovian queueing model; perturbed four-banded matrix; spectral decomposition; Toeplitz solver
Mathematics Subject Classification: 15A05, 15B05, 65T50

1. Introduction

Since Erlang proposed the idea of queueing theory [1], it has been widely used in telecommunication and other fields. There are various queueing models due to different arrival processes, waiting lists, queueing disciplines, and service processes, such as $M/G/1, GI/M/1$. Consider an $M^{[x]}/M/1/n-1$ queue model with feedback for a first-come-first-serve rule [2]. The queue has a Poisson batch arrival of requests (mean batch interarrival time $\frac{1}{b}$), exponential service times (mean $\frac{1}{u}$), a single server, and finite waiting rooms of size $n-1$. Let the maximum batch size be 2 and, f_1 and f_2 denote the probability of the group size 1 and 2, respectively, where $f_1 + f_2 = 1$, and $c = f_1 b$, $d = f_2 b$. Let h be the probability that a customer is fed back immediately after his service completion and $a = (1-h)u$. The infinitesimal generator $\mathfrak{P} = (p_{i,j})_{i,j=1}^{n}$ of the traffic process associated with the Markovian queueing model [2] is the band matrix with lower bandwidth 1 and upper bandwidth 2, where

$$p_{i,j} = \begin{cases} -b, & i = j = 1, \\ -(a+b), & i = j = 2, 3, \cdots, n-2, \\ -(a+c+ud), & i = j = n-1, \\ -a, & i = j = n, \\ a, & i = j+1, j = 1, 2, \cdots, n-1, \\ c, & i = j-1, j = 2, 3, \cdots, n-1, \\ c+ud, & i = j-1, j = n, \\ d, & i = j-2, j = 3, 4, \cdots, n, \\ 0, & \text{otherwise}. \end{cases} \tag{1.1}$$

The infinitesimal generator is an important index for analyzing the queueing model. The steady-state probability distribution in queuing models is typically obtained by solving a system of linear equations. Oda [2] utilized the infinitesimal generator, alongside the steady-state distribution and the rate matrices of the traffic process, to formulate the recursive linear equations necessary for calculating the moments of the traffic process. In contrast to earlier studies [3], Oda's methodology can be extended to a broader range of applications due to the sparsity and unique structure of the infinitesimal generator. Additionally, Wen et al. [4] utilized a multipreconditioned generalized minimal residual (GMRES) method, incorporating multiple preprocessors, to solve linear systems associated with generation matrices, thereby deriving the steady-state probability distribution of stochastic automatic networks within the context of queueing system modeling.

In this paper, by perturbing the first row of the infinitesimal generator of the above traffic process associated with the Markovian queueing model, we study the perturbed four-banded linear system

$$\mathfrak{G}x = y, \tag{1.2}$$

where $\mathfrak{G} = (g_{i,j})_{i,j=1}^{n}$, the entries $g_{i,j}$ are the same as $p_{i,j}$ in Eq (1.1), except for $g_{1,n} = 0$, $x = (x_1, x_2, ..., x_n)^{\mathrm{T}}$, and $y = (y_1, y_2, ..., y_n)^{\mathrm{T}}$. From the perspective of structure, the matrix $\mathfrak{G}$ is a perturbed banded Toeplitz matrix.

Early research on the solution of banded linear systems predominantly utilized direct methods and iterative methods. While these approaches demonstrated efficacy for small-scale problems, their computational complexity and storage requirements escalated. In recent years, with in-depth research into the structural properties of matrices, numerous efficient algorithms have been developed specifically for structured matrices, such as fast Fourier transform (FFT)-based methods [5–7], Krylov subspace methods [8], preconditioning techniques [9], parallel computing and GPU acceleration [10], and so on [11–15]. For example, Bini and Meini [5] integrated the techniques of cyclic reduction and displacement rank to formulate an efficient algorithm with a computational complexity of $O(n \log n)$ for solving banded Toeplitz systems. Building upon the fast Fourier transform, Fischer et al. [6] systematically explored the use of Toeplitz factorizations in conjunction with the Sherman-Morrison-Woodbury (SMW) formula for solving symmetric banded Toeplitz linear systems. Subsequently, Malyshev and Sadkane [7] advanced this methodology by integrating spectral factorization of generating functions with the SMW formula as an alternative to cyclic reduction, thereby extending Fischer et al.'s [6] approach to accommodate large-scale nonsymmetric systems.

On the other hand, preconditioners are crucial for accelerating the convergence of iterative methods. Serra Capizzano and Tablino Possio [9] introduced a multigrid technique employing multilevel circulant matrices as preprocessors to efficiently handle the banded systems arising from the discretization of partial differential equations. This approach significantly reduces computational complexity to a linear scale of $O(n)$, encompassing $O(n)$ arithmetic operations and $O(1)$ memory operations. In addition, utilizing GPU and distributed computing technology, some scholars have proposed parallel algorithms to further accelerate the solution of large-scale perturbed banded systems. Jandron et al. [10] introduced a novel methodology for solving banded linear systems utilizing asynchronous direct solvers, where the key mechanism is that reduction to row-echelon form is not required by the solver. However, in distributed computing, data communication between nodes may become a performance bottleneck, especially in sparse linear systems.

In certain low-rank perturbed linear systems, some researchers employ low-rank decomposition techniques to convert high-dimensional problems into low-dimensional ones, thereby substantially reducing computational complexity [16–23]. Sogabe [24] decomposed the coefficient matrix in comrade linear systems into a tridiagonal matrix and a low-rank matrix. Jia and Li [25] gave two algorithms for solving opposite-bordered tridiagonal systems based on matrix decomposition. In accordance with the structural properties of the coefficient matrix discussed in this paper, a divide-and-conquer strategy is employed. This approach involves decomposing the large-scale problem into smaller subproblems via matrix decomposition, thereby enhancing computational efficiency. Following two kinds of decompositions, two distinct types of sublinear systems are identified: Toeplitz systems and four-banded linear systems. By leveraging existing Toeplitz processors, a solution method with $O(n \log n)$ complexity can be realized. Unlike most iterative methods, direct methods are generally straightforward to implement. For banded linear systems, the band LU decomposition can be directly applied to achieve a solution method with linear complexity. Specifically, it has a cost of $O(n)$ arithmetic operations with $O(n)$ memory requirements during the formation of the U factor in the LU decomposition [26].

The rest of this paper is organized as follows. In Section 2, we introduce a decomposition method for the linear system that utilizes a Toeplitz solver, followed by a discussion of a decomposition method based on band LU decomposition in Section 3. In Section 4, we provide several numerical examples to demonstrate the effectiveness of these methods. Finally, Section 5 concludes the paper.

2. Toeplitz solver-based algorithm for solving a perturbed four-banded linear system

This section is organized into two subsections. The first subsection introduces the decomposition form based on matrix structure, while the second subsection discusses the Toeplitz linear solver.

2.1. Toeplitz solver-based algorithm

Observing the structure of the perturbed four-banded matrix $\mathfrak{G}$, we can separate it into the sum of a four-banded Toeplitz matrix and three rank-one matrices. Examining perturbed elements from various perspectives, such as columns or rows, results in different methods for decomposing the matrix $\mathfrak{G}$. We focus exclusively on column decomposition to illustrate the decomposition algorithm. Analogously, readers are encouraged to develop their own decomposition algorithms for row decomposition, following the principles demonstrated in the column decomposition example.

After separation, we need to combine the SMW formula to calculate the inverse matrix of the decomposed matrices. However, to know the conditioning of the decomposed matrix is essential for reliable numerical results in terms of precision [27–30]. Furthermore, the SMW can be unstable when ill-conditioning is present [26]. In order to ensure the stability of the SMW, we explore strictly diagonally dominant linear systems and decomposed linear systems in this paper. Thus, we translate the problem of solving the linear system $\mathfrak{G}x = y$ into four four-banded Toeplitz linear systems.

For convenience to describe our algorithm, we give the SMW formula and the inverse of identity matrix plus three rank-one matrices below.

Lemma 2.1. ([31], Generalized Sherman-Morrison-Woodbury Formula) *Let A be an n-by-n matrix. Let U_k and V_k be matrices of size n-by-m. Then*

$$\left(A + \sum_{k=1}^{n} U_k V_k^{\mathrm{T}}\right)^{-1} = A^{-1} - A^{-1}[U_1, U_2, \ldots, U_n]M^{-1}\begin{bmatrix} V_1^{\mathrm{T}} \\ V_2^{\mathrm{T}} \\ \vdots \\ V_n^{\mathrm{T}} \end{bmatrix} A^{-1},$$

where M is an nm-by-nm matrix given by

$$M = \begin{bmatrix} I_m + V_1^{\mathrm{T}}A^{-1}U_1 & V_1^{\mathrm{T}}A^{-1}U_2 & \cdots & V_1^{\mathrm{T}}A^{-1}U_n \\ V_2^{\mathrm{T}}A^{-1}U_1 & I_m + V_2^{\mathrm{T}}A^{-1}U_2 & \cdots & V_2^{\mathrm{T}}A^{-1}U_n \\ \vdots & \vdots & \ddots & \vdots \\ V_n^{\mathrm{T}}A^{-1}U_1 & V_n^{\mathrm{T}}A^{-1}U_2 & \cdots & I_m + V_n^{\mathrm{T}}A^{-1}U_n \end{bmatrix}.$$

Corollary 2.1. *Let $u_1, u_2, u_3, v_1, v_2, v_3$ be vectors of length n. Then, the inverse of the matrix of the form $I_n + u_1 v_1^{\mathrm{T}} + u_2 v_2^{\mathrm{T}} + u_3 v_3^{\mathrm{T}}$ can be explicitly given by*

$$(I_n + u_1 v_1^{\mathrm{T}} + u_2 v_2^{\mathrm{T}} + u_3 v_3^{\mathrm{T}})^{-1} = I_n - [u_1,\ u_2,\ u_3]M^{-1}\begin{bmatrix} v_1^{\mathrm{T}} \\ v_2^{\mathrm{T}} \\ v_3^{\mathrm{T}} \end{bmatrix},$$

where

$$M = \begin{bmatrix} 1 + v_1^{\mathrm{T}}u_1 & v_1^{\mathrm{T}}u_2 & v_1^{\mathrm{T}}u_3 \\ v_2^{\mathrm{T}}u_1 & 1 + v_2^{\mathrm{T}}u_2 & v_2^{\mathrm{T}}u_3 \\ v_3^{\mathrm{T}}u_1 & v_3^{\mathrm{T}}u_2 & 1 + v_3^{\mathrm{T}}u_3 \end{bmatrix}.$$

For the convenience of describing the theorem, some notation explanations are given below. $\mathfrak{Q} = (q_{i,j})_{i,j=1}^{n}$ is a four-banded Toeplitz matrix with lower bandwidth 1 and upper bandwidth 2, where $q_{i,i} = -(a+b)(i = 1, 2, \cdots, n)$, $q_{i,i-1} = a(i = 2, 3, \cdots, n)$, $q_{i,i+1} = c(i = 1, 2, \cdots, n-1)$, $q_{i,i+2} = d(i = 1, 2, \cdots, n-2)$, and otherwise, $q_{i,j} = 0$.

Theorem 2.1. *Consider the strictly diagonally dominant perturbed four-banded linear system* $\mathfrak{G}x = y$ *as shown in Eq (1.2). Then*

$$x = \phi - \frac{r}{m}\mu - \frac{s}{m}\nu - \frac{t}{m}\omega, \tag{2.1}$$

where $\phi = [\phi_1, \phi_2, \cdots, \phi_n]^{\mathrm{T}}$, $\mu = [\mu_1, \mu_2, \cdots, \mu_n]^{\mathrm{T}}, \nu = [\nu_1, \nu_2, \cdots, \nu_n]^{\mathrm{T}}$, *and* $\omega = [\omega_1, \omega_2, \cdots, \omega_n]^{\mathrm{T}}$ *are the solution of* $\mathfrak{Q}x = y$, $\mathfrak{Q}x = \xi$, $\mathfrak{Q}x = \delta$, *and* $\mathfrak{Q}x = \varphi$, *respectively, and* $\mathfrak{Q}$ *is a non-singular four-banded matrix,*

$$\begin{aligned} m = &\ (1+\mu_1)(1+\nu_{n-1})(1+\omega_n) + \mu_n\nu_1\omega_n + \mu_{n-1}\nu_n\omega_{n-1} - \mu_n(1+\nu_{n-1})\omega_1 - \mu_{n-1}\nu_1(1+\omega_n) - \\ &\ (1+\mu_1)\nu_n\omega_{n-1}, \end{aligned} \tag{2.2}$$

$$r = [(1+\nu_{n-1})(1+\omega_n) - \nu_n\omega_{n-1}]\phi_1 + [\nu_n\omega_1 - \nu_1(1+\omega_n)]\phi_{n-1} + [\nu_1\omega_{n-1} - (1+\nu_{n-1})\omega_1]\phi_n, \tag{2.3}$$

$$s = [\mu_n\omega_{n-1} - \mu_{n-1}(1+\omega_n)]\phi_1 + [(1+\mu_1)(1+\omega_n) - \mu_n\omega_1]\phi_{n-1} + [\mu_{n-1}\omega_1 - (1+\mu_1)\omega_{n-1}]\phi_n, \tag{2.4}$$

$$t = [\mu_{n-1}\nu_n - \mu_n(1+\nu_{n-1})]\phi_1 + [\mu_n\nu_1 - (1+\mu_1)\nu_n]\phi_{n-1} + [(1+\mu_1)(1+\nu_{n-1}) - \mu_{n-1}\nu_1]\phi_n. \tag{2.5}$$

Proof. Consider column element perturbations of the matrix $\mathfrak{G}$. Then, the matrix $\mathfrak{G}$ has the following decomposition:

$$\mathfrak{G} = \mathfrak{Q} + \xi e_1^{\mathrm{T}} + \delta e_{n-1}^{\mathrm{T}} + \varphi e_n^{\mathrm{T}},$$

where $\xi = [a, 0, \cdots, 0]^{\mathrm{T}}$, $\delta = [0, \cdots, 0,\ b-c-ud, 0]^{\mathrm{T}}$, $\varphi = [-a, 0, \cdots, 0,\ ud,\ b]^{\mathrm{T}}$, and $e_i (i = 1, n-1, n)$ is the ith unit vector. Assume that $\mathfrak{Q}$ is a nonsingular matrix.

Then the linear system $\mathfrak{G}x = y$ satisfies

$$\mathfrak{Q}(I_n + \mathfrak{Q}^{-1}\xi e_1^{\mathrm{T}} + \mathfrak{Q}^{-1}\delta e_{n-1}^{\mathrm{T}} + \mathfrak{Q}^{-1}\varphi e_n^{\mathrm{T}})x = y, \tag{2.6}$$

where I_n is the n-by-n identity matrix.

Let $\mathfrak{Q}^{-1}\xi = \mu$, $\mathfrak{Q}^{-1}\delta = \nu$, $\mathfrak{Q}^{-1}\varphi = \omega$, and $\mathfrak{Q}^{-1}y = \phi$, where $\mu = [\mu_1, \mu_2, \cdots, \mu_n]^{\mathrm{T}}, \nu = [\nu_1, \nu_2, \cdots, \nu_n]^{\mathrm{T}}$, $\omega = [\omega_1, \omega_2, \cdots, \omega_n]^{\mathrm{T}}$, and $\phi = [\phi_1, \phi_2, \cdots, \phi_n]^{\mathrm{T}}$, respectively. In this case, Eq (2.6) is equivalent to the following equation:

$$(I_n + \mu e_1^{\mathrm{T}} + \nu e_{n-1}^{\mathrm{T}} + \omega e_n^{\mathrm{T}})x = \phi.$$

Calculate the above equation, and we have

$$x = (I_n + \mu e_1^{\mathrm{T}} + \nu e_{n-1}^{\mathrm{T}} + \omega e_n^{\mathrm{T}})^{-1}\phi. \tag{2.7}$$

According to Corollary 2.1, we have

$$(I_n + \mu e_1^{\mathrm{T}} + \nu e_{n-1}^{\mathrm{T}} + \omega e_n^{\mathrm{T}})^{-1} = I_n - \begin{pmatrix} \mu & \nu & \omega \end{pmatrix} \cdot \begin{pmatrix} 1+\mu_1 & \nu_1 & \omega_1 \\ \mu_{n-1} & 1+\nu_{n-1} & \omega_{n-1} \\ \mu_n & \nu_n & 1+\omega_n \end{pmatrix}^{-1} \cdot \begin{pmatrix} e_1^{\mathrm{T}} \\ e_{n-1}^{\mathrm{T}} \\ e_n^{\mathrm{T}} \end{pmatrix}.$$

Substitute the above results into Eq (2.7) to obtain

$$x = \phi - \begin{pmatrix} \mu & \nu & \omega \end{pmatrix} \cdot \begin{pmatrix} 1+\mu_1 & \nu_1 & \omega_1 \\ \mu_{n-1} & 1+\nu_{n-1} & \omega_{n-1} \\ \mu_n & \nu_n & 1+\omega_n \end{pmatrix}^{-1} \cdot \begin{pmatrix} \phi_1 \\ \phi_{n-1} \\ \phi_n \end{pmatrix}$$

$$= \phi - \frac{r}{m}\mu - \frac{s}{m}\nu - \frac{t}{m}\omega,$$

where m, r, s, t are the same as in Eqs (2.2)–(2.5). □

Based on the above Theorem 2.1, we give the following Algorithm 1 for solving the linear system $\mathfrak{G}x = y$.

Algorithm 1 Toeplitz solver-based (TS-based) algorithm for solving $\mathfrak{G}x = y$.

Step 1: Solve $\mathfrak{Q}\mu = \xi$, $\mathfrak{Q}\nu = \delta$, $\mathfrak{Q}\omega = \varphi$, and $\mathfrak{Q}\phi = y$ by any Toeplitz linear solver.

Step 2: Determine the value of m, r, s, t by Eqs (2.2)–(2.5) and calculate $x = \phi - \frac{r}{m}\mu - \frac{s}{m}\nu - \frac{t}{m}\omega$ by Eq (2.1).

From the above TS-based algorithm, it can be seen that we need to solve a Toeplitz linear system with multiple right-hand sides. We can see that the amount of calculation and storage in the second step of the TS-based algorithm are both $O(n)$. The main work of the algorithm is to solve Toeplitz linear systems, such as $\mathfrak{Q}\mu = \xi$. One method for solving Toeplitz linear systems involves combining the FFT algorithm, which reduces the time complexity to $O(n \log n)$. In addition, there are also some iterative methods, such as the generalized minimum residual method [8, 32–35]. Even for ill-conditioned Toeplitz systems, Chan et al. [36] gave a solution by constructing the new circulant preconditioners. Overall, the complexity of the TS-based algorithm is $O(n \log n)$.

Next, we introduce some Toeplitz linear solvers to support the algorithm mentioned above.

2.2. Toeplitz linear solver

Lemma 2.2. *[37] Let $T = (t_{i-j})_{i,j=1}^{n} \in C^{n\times n}$ be a Toeplitz matrix. If the vectors $u = [u_1, u_2, \cdots, u_n]^T$, $v = [v_1, v_2, \cdots, v_n]^T$, e_1, and e_n are the first and last unit vectors satisfying*

$$Tu = e_1, \ Tv = e_n, \tag{2.8}$$

and $u_1 \neq 0$, then T is invertible and

$$T^{-1} = -\frac{1}{2u_1}(C_1 S_1 - C_2 S_2), \tag{2.9}$$

where C_1, C_2 are circulant matrices with u, $\ddot{v} = [v_n, v_1, v_2, \cdots, v_{n-1}]^T$ is their first columns, respectively; S_1, S_2 are skew-circulant matrices with $\dot{v} = [-v_n, v_1, v_2, \cdots, v_{n-1}]^T$, and u is their first column, respectively.

Perform spectral decomposition on the circulant matrix and skew-circulant matrix in Eq (2.9), and then Eq (2.9) can be rewritten as

$$T^{-1} = -\frac{1}{2u_1}F^*(\Lambda_1 F\Omega^* F^* \Lambda_2 - \Lambda_3 F\Omega^* F^* \Lambda_4)F\Omega, \tag{2.10}$$

where the Fourier matrix $F = (F_{j,k})_{j,k=1}^{n}$, $F_{j,k} = \frac{1}{\sqrt{n}}e^{\frac{2\pi i(j-1)(k-1)}{n}}$, $1 \leq j, k \leq n$, $\Omega = \mathrm{diag}(1, e^{\frac{-i\pi}{n}}, \ldots, e^{\frac{-i(n-1)\pi}{n}})$, F^*, Ω^* represent the conjugate transpose of F, Ω, respectively, and $\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4$ are diagonal matrices containing the eigenvalues of C_1, S_1, C_2, S_2, respectively [38].

Algorithm 2 Compute $p = T^{-1}q$.

Step 1: $p_1 = F\Omega q$;
Step 2: $p_2 = \Lambda_1 F\Omega^* F^* \Lambda_2 p_1$;
Step 3: $p_3 = \Lambda_3 F\Omega^* F^* \Lambda_4 p_1$;
Step 4: $p = -\frac{1}{2u_1} F^*(p_2 - p_3)$.

Based on the representation of Toeplitz matrix inversion, we give the following Algorithm 2 for computing the product of Toeplitz matrix inversion T^{-1} and a vector q.

The main workload of Algorithm 2 can be done by three FFTs and three inverse FFTs (IFFTs), more precisely, $9n \log n + O(n)$. Besides, it can be seen from the above description that the first step needs to solve two Toeplitz systems in Eq (2.8) and then obtain $\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4, u_1$. For high-order linear systems with Toeplitz structure, the Krylov subspace method has certain advantages. In addition, because the condition number of the Toeplitz matrix in general is not very small, Lei and Huang proposed the generalized minimal residual (GMRES) method with a Strang-type block-circulant preconditioner to accelerate the convergence of the GMRES [35]. Here, we use the preconditioned GMRES (PGMRES) to solve two Toeplitz systems in Eq (2.8). The preparatory work for Algorithm 2 is given by Algorithm 3 below.

Algorithm 3 Compute $\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4, u_1$.

Step 1: Solve $Tu = e_1$ by the PGMRES method;
Step 2: Solve $Tv = e_n$ by the PGMRES method;
Step 3: $\dot{v} = [-v_n, v_1, v_2, \cdots, v_{n-1}]^T$;
Step 4: $\ddot{v} = [v_n, v_1, v_2, \cdots, v_{n-1}]^T$;
Step 5: $\Lambda_1 = Fu$;
Step 6: $\Lambda_2 = F\Omega\dot{v}$;
Step 7: $\Lambda_3 = F\ddot{v}$;
Step 8: $\Lambda_4 = F\Omega u$.

The complexity of solving the two linear systems can be regarded as $O(n \log n)$ since the iteration number of the PGMRES method is usually very small [32]. The workload of computing $\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4$ is four FFTs, i.e., $6n \log n + O(n)$. Overall, the complexity of this Toeplitz linear solver is $O(n \log n)$.

3. The band LU decomposition-based algorithm for solving a perturbed four-banded linear system

In this section, we give the algorithm for solving a perturbed four-banded linear system based on band LU decomposition. Unlike the previous section, we separate the matrix $\mathfrak{G}$ into the sum of a four-banded matrix and one rank-one matrix. For the sake of convenience in describing our algorithm, and considering the varying numbers of rank-one matrices, we present a corollary of the SMW formula, the inverse of the identity matrix plus one rank-one matrix.

Corollary 3.1. *Let u_1 and v_1 be vectors of length n. Then, the inverse of the matrix of the form $I_n + u_1 v_1^{\mathrm{T}}$*

can be explicitly given by

$$(I_n + u_1 v_1^{\mathrm{T}})^{-1} = I_n - \frac{u_1 v_1^{\mathrm{T}}}{1 + v_1^{\mathrm{T}} u_1}.$$

Theorem 3.1. *Consider the strictly diagonally dominant perturbed four-banded linear system $\mathfrak{G}x = y$ as shown in Eq (1.2). Then*

$$x = \psi - \frac{\psi_n}{1 + \tau_n}\tau, \tag{3.1}$$

where ψ and τ are the solutions of $\mathfrak{P}x = y$ and $\mathfrak{P}x = \eta$, respectively, and $\mathfrak{P}$ is a non-singular four-banded matrix and the same as in Eq (1.1).

Proof. Consider column element perturbations of the matrix $\mathfrak{G}$. Then, the matrix $\mathfrak{G}$ has the following decomposition:

$$\mathfrak{G} = \mathfrak{P} + \eta e_n^{\mathrm{T}},$$

where $\eta = [-a, 0, \cdots, 0, 0]^{\mathrm{T}}$, e_n is the nth unit vector, and $\mathfrak{P}$ is a non-singular four-banded matrix and the same as in Eq (1.1). Here, we assume $\mathfrak{P}$ is a nonsingular matrix.

Then the linear system $\mathfrak{G}x = y$ satisfies

$$\mathfrak{P}(I_n + \mathfrak{P}^{-1}\eta e_n^{\mathrm{T}})x = y, \tag{3.2}$$

where I_n is the n-by-n identity matrix. Let $\mathfrak{P}^{-1}\eta = \tau$ and $\mathfrak{P}^{-1}y = \psi$, where $\tau = [\tau_1, \tau_2, \cdots, \tau_n]^{\mathrm{T}}$ and $\psi = [\psi_1, \psi_2, \cdots, \psi_n]^{\mathrm{T}}$, respectively. In this case, Eq (3.2) is equivalent to the following equation:

$$(I_n + \tau e_n^{\mathrm{T}})x = \psi.$$

Calculate the above equation, and we have

$$x = (I_n + \tau e_n^{\mathrm{T}})^{-1}\psi. \tag{3.3}$$

According to Corollary 3.1, we have

$$(I_n + \tau e_n^{\mathrm{T}})^{-1} = I_n - \frac{\tau e_n^{\mathrm{T}}}{1 + \tau_n}.$$

Substitute the above results into Eq (3.3) to obtain

$$x = (I_n - \frac{\tau e_n^{\mathrm{T}}}{1 + \tau_n})\psi.$$

Subsequent simplification yields the result

$$x = \psi - \frac{\psi_n}{1 + \tau_n}\tau.$$

□

Based on the above Theorem 3.1, we give the following Algorithm 4 for solving the linear system $\mathfrak{G}x = y$.

Algorithm 4 Band LU decomposition-based (BLUD-based) algorithm for solving $\mathfrak{G}x = y$.

Step 1: Solve $\mathfrak{P}\psi = y$ and $\mathfrak{P}\tau = \eta$ by band LU decomposition.

Step 2: Calculate $x = \psi - \frac{\psi_n}{1+\tau_n}\tau$ by Eq (3.1).

From the above BLUD-based algorithm for solving the linear system $\mathfrak{G}x = y$, the main work is to solve the four-banded linear sysytem with lower bandwidth 1 and upper bandwidth 2. The number of floating-point operations (flops) required to solve an n-by-n banded linear system using the band LU decomposition method is $O(n)$ [26]. It can be seen that the number of flops in the second step can be basically ignored. Overall, the complexity of the BLUD-based algorithm is $O(n)$.

4. Numerical experiments

In the experiments, to test the performance of the proposed algorithms for perturbed four-banded linear system $\mathfrak{G}x = y$, we run two examples using the TS-based algorithm, BLUD-based algorithm, and backslash operator (MATLAB). From the analysis in the previous section, we know that the TS-based algorithm is based on the existing Toeplitz solvers. Here we use the Toeplitz solver based on Algorithms 2 and 3 in Section 2.2. For the PGMRES method, the stopping criterion we used is $\frac{\|r^k\|_2}{\|r^0\|_2} < 10^{-14}$, where r^k is the residual vector of the linear system after k iterations, the restart iteration we used is 500, and the initial guess is chosen as the zero vector. This Toeplitz solver is dealing with strictly diagonally dominant linear systems, so the coefficient matrices of these systems must be guaranteed to be strictly diagonally dominant. In order to ensure the stability of the SMW, these decomposed linear systems are also strictly diagonally dominant linear systems in this section. Here, we calculate the condition number of the coefficient matrix for each algorithm to demonstrate the stability of the SMW formula.

All experiments are performed in MATLAB R2022a on DESKTOP-CPVUBT4 with Intel (R) Core (TM) i7-13700k CPU 3.40 GHz and 32GB RAM. In the following tables, "n" denotes the matrix order, "Cond" denotes the condition number of the coefficient matrices $\mathfrak{G}$, $\mathfrak{Q}$, and $\mathfrak{P}$, "RelError = $\frac{\|x-x_{\text{exact}}\|}{\|x_{\text{exact}}\|}$" denotes the relative error of x, where x_{exact} represents the exact solution, "Residual = $\|\mathfrak{G}x - y\|$" means the residual error, "Time(s)" is the total CPU time in seconds, and "NaN" represents calculation failure due to a very large matrix and insufficient memory or computing resources. 100 repeated simulations are conducted for these examples.

Example 4.1. *Consider an n-by-n perturbed four-banded linear system* $\mathfrak{G}x = y$ *where* a, b, c, d, u *are random numbers generated from open interval* $(0, 0.1)$. *Set* $x_{exact} = [1, 1, ..., 1]^{\mathrm{T}}$ *and* $y = \mathfrak{G}x_{exact}$.

Table 1 shows the results of three methods for perturbed four-banded linear systems in terms of relative error and computation time in Example 4.1. From the selected data, it can be seen that the relative errors of the three algorithms are not significantly different, and all remain at least 10^{-14}.

Table 1. Comparison of the relative error and computation time for Example 4.1.

n	Backslash operator			BLUD-based algorithm			TS-based algorithm		
	Cond	RelError	Time(s)	Cond	RelError	Time(s)	Cond	RelError	Time(s)
2^6	32.7778	1.1832e-16	3.2618e-05	14.4002	1.9611e-16	0.0011	7.1748	4.9257e-15	0.0024
2^8	16.6655	2.8159e-16	1.7969e-04	13.7407	2.2611e-16	0.0050	4.0125	1.5297e-14	0.0028
2^{10}	34.2421	1.1832e-17	6.7969e-04	34.3333	1.5297e-17	0.0161	4.8849	4.2611e-15	0.0032
2^{12}	18.8910	9.8473e-17	0.0035	10.8810	4.4473e-17	0.1370	2.4620	3.3609e-14	0.0544
2^{14}	37.1825	1.8154e-17	0.0147	25.3417	1.1154e-16	0.5062	5.1873	1.5947e-14	0.2025
2^{16}	57.8503	3.8173e-17	0.0639	40.7000	2.3726e-16	1.9346	8.5577	2.4790e-14	0.8423
2^{18}	60.1530	3.8173e-17	0.2431	36.9918	2.3726e-16	7.1146	10.3718	2.4790e-14	3.4423
2^{20}	NaN			NaN			NaN		

Regarding computational efficiency, the backslash operator in MATLAB demonstrates superior performance. Furthermore, the BLUD-based algorithm outperforms the TS-based algorithm for linear systems of order less than 2^8. However, as the system order increases, this advantage diminishes, contradicting the theoretical analysis presented in Sections 2 and 3. This discrepancy is primarily attributable to the matrix operations and storage demands inherent in the internal elimination and replacement processes of the BLUD-based algorithm.

Upon further observation of the condition number of the coefficient matrix, it is evident that the backslash operator (MATLAB) has the highest condition number, followed by the BLUD-based algorithm, and the TS-based algorithm has the lowest. We know that the larger the condition number, the worse the numerical stability of the matrix. When the matrix order is greater than 2^{20}, it may indicate a calculation failure due to a very large matrix and insufficient memory or computing resources. Therefore, we can say that although the algorithms we provide do not have as good computation time as the backslash operator (MATLAB), they are more stable.

Example 4.2. *Consider an n-by-n perturbed four-banded linear system* $\mathfrak{G}x = y$ *where* a, b, c, d, u, y *are random numbers.*

Table 2 shows the results of three methods in residual and computation time for the perturbed four-banded linear system in Example 4.2. Similar to Example 4.1, regarding the computation time, the backslash operator (MATLAB) demonstrates superior performance. As the matrix order increases, the TS-based algorithm performs well, and the advantages of the TS-based algorithm become greater. In terms of condition numbers, the TS-based algorithm performs the best. Therefore, the TS-based algorithm has a good comprehensive performance ability.

Table 2. Comparison of the residual error and computation time for Example 4.2.

n	Backslash operator			BLUD-based algorithm			TS-based algorithm		
	Cond	Residual	Time(s)	Cond	Residual	Time(s)	Cond	Residual	Time(s)
2^6	20.5933	6.5983e-15	9.2829e-05	11.3683	6.9723e-15	0.0011	4.3527	4.8137e-14	0.0040
2^8	20.4567	2.1479e-15	1.2969e-04	13.2324	2.3916e-15	0.0042	3.7568	9.4897e-14	0.0059
2^{10}	36.6288	4.8832e-15	7.1319e-04	26.2567	6.8297e-15	0.0183	5.8124	1.2611e-14	0.0132
2^{12}	40.2399	3.8444e-14	0.0035	19.3468	3.4473e-14	0.0724	8.9512	3.3609e-13	0.0194
2^{14}	29.2466	6.9394e-14	0.0135	18.3458	7.0154e-14	0.2796	5.5572	2.8317e-13	0.0867
2^{16}	21.7949	1.6245e-14	0.0522	15.8294	7.1208e-14	1.1215	3.4736	2.4790e-12	0.2996
2^{18}	22.8314	6.6724e-14	0.2553	11.8327	8.9776e-14	4.1146	4.4563	7.4581e-12	1.8524
2^{20}	NaN			NaN			NaN		

5. Conclusions

This paper investigates the perturbed four-banded linear system derived from the traffic process associated with a Markovian queueing model—$M^{[x]}/M/1/n-1$—with feedback for a first-come-first-serve rule. Initially, the perturbed four-banded matrix is decomposed into the sum of a four-banded Toeplitz matrix and three rank-one matrices. Utilizing the spectral decomposition of circulant and skew circulant matrices, we compute the product of the Toeplitz inversion and a vector. Additionally, the perturbed four-banded matrix is further decomposed into the sum of a four-banded matrix and a rank-one matrix. Band LU decomposition is then applied to solve the resulting separated linear systems. To evaluate the performance of the proposed algorithms, we conduct numerical simulations comparing computation time, relative error, and residuals of test cases against the backslash operator in MATLAB. The results indicate that the proposed algorithms demonstrate favorable performance in terms of computational efficiency and stability.

Author contributions

Jiaqi Qu: Writing-original draft, Writing-review & editing; Yanpeng Zheng: Conceptualization, Visualization, Writing-review & editing; Yunlan Wei and Zhaolin Jiang: Conceptualization, Funding acquisition, Visualization, Writing-review & editing. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

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

Acknowledgments

This work is partially supported by the Natural Science Foundation of Shandong Province (Grant Nos. ZR2022QA057 and ZR2022MA092). This support has been instrumental in facilitating the research activities and outcomes presented in this work.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. G. Giambene, *Queuing theory and telecommunications networks and applications*, 2 Eds., New York: Springer, 2005. http://dx.doi.org/10.1007/978-1-4614-4084-0
2. T. Oda, Moment analysis for traffic associated with Markovian queueing systems, *IEEE T. Commun.*, **39** (1991), 737–746. http://dx.doi.org/10.1109/26.87164
3. M. Fumiaki, An infinitely many server queue having Markov renewal arrivals and hyperexponential service times, *J. Oper. Res. Soc. Jpn.*, **29** (2017), 338–351. http://dx.doi.org/10.2307/2582311
4. C. Wen, T. Z. Huang, X. M. Gu, Z. L. Shen, H. F. Zhang, C. Liu, Multipreconditioned GMRES for simulating stochastic automata networks, *Open Math.*, **16** (2018), 986–998. http://dx.doi.org/10.1515/math-2018-0083
5. D. A. Bini, B. Meini, Effective methods for solving banded Toeplitz systems, *SIAM J. Numer. Anal.*, **20** (1999), 700–719. http://dx.doi.org/10.1137/S0895479897324585
6. D. Fischer, G. Golub, O. Hald, C. Leiva, O. Widlund, On Fourier-Toeplitz methods for separable elliptic problems, *Math. Comput.*, **28** (1974), 349–368. http://dx.doi.org/10.2307/2005913
7. A. N. Malyshev, M. Sadkane, Fast solution of unsymmetric banded Toeplitz systems by means of spectral factorizations and Woodbury's formula, *Numer. Linear Algebra Appl.*, **21** (2014), 13–23. https://doi.org/10.1002/nla.1853
8. Y. L. Zhao, P. Y. Zhu, X. M. Gu, X. L. Zhao, J. Cao, A limited-memory block bi-diagonal Toeplitz preconditioner for block lower triangular Toeplitz system from time-space fractional diffusion equation, *J. Comput. Appl. Math.*, **362** (2019), 99–115. http://dx.doi.org/10.1016/j.cam.2019.05.019
9. S. Serra-Capizzano, C. Tablino-Possio, Multigrid methods for multilevel circulant matrices, *SIAM J. Sci. Comput.*, **26** (2004), 55–85. http://dx.doi.org/10.1137/S1064827501388509
10. M. A. Jandron, A. A. Ruffa, J. Baglama, An asynchronous direct solver for banded linear systems, *Numer. Algorithms*, **76** (2017), 211–235. http://doi.org/10.1007/s11075-016-0251-3
11. Y. R. Fu, X. Y. Jiang, Z. L. Jiang, S. Jhang, Fast algorithms for finding the solution of CUPL-Toeplitz linear system from Markov chain, *Appl. Math. Comput.*, **396** (2021), 125859. http://doi.org/10.1016/j.amc.2020.125859

12. X. Zhang, X. Y. Jiang, Z. L. Jiang, H. Byun, An improvement of methods for solving the CUPL-Toeplitz linear system, *Appl. Math. Comput.*, **421** (2022), 126932. http://doi.org/10.1016/j.amc.2022.126932

13. X. Zhang, Y. P. Zheng, Z. L. Jiang, H. Byun, Numerical algorithms for corner-modified symmetric Toeplitz linear system with applications to image encryption and decryption, *J. Appl. Math. Comput.*, **69** (2023), 1967–1987. http://doi.org/10.1007/s12190-022-01819-7

14. X. Zhang, Y. P. Zheng, Z. L. Jiang, H. Byun, Fast algorithms for perturbed Toeplitz-plus-Hankel system based on discrete cosine transform and their applications, *Jpn. J. Ind. Appl. Math.*, **41** (2024), 567–583. http://doi.org/10.1007/s13160-023-00616-4

15. X. Zhang, Y. P. Zheng, Z. L. Jiang, Fast algorithms for the solution of perturbed symmetric Toeplitz linear system and its applications, *Comput. Appl. Math.*, **43** (2024), 252. http://dx.doi.org/10.1007/s40314-024-02773-9

16. J. T. Jia, S. M. Li, On the inverse and determinant of general bordered tridiagonal matrices, *Comput. Math. Appl.*, **69** (2015), 503–509. http://dx.doi.org/10.1016/j.camwa.2015.01.012

17. J. T. Jia, T. Sogabe, S. M. Li, A generalized symbolic Thomas algorithm for the solution of opposite-bordered tridiagonal linear systems, *J. Comput. Appl. Math.*, **290** (2015), 423–432. http://dx.doi.org/10.1016/j.cam.2015.05.026

18. J. A. Marrero, A numerical solver for general bordered tridiagonal matrix equations, *Comput. Math. Appl.*, **72** (2016), 2731–2740. http://dx.doi.org/10.1016/j.camwa.2016.09.025

19. A. Martin, I. D. Boyd, Variant of the Thomas Algorithm for opposite-bordered tridiagonal systems of equations, *Int. J. Numer. Meth. Bio.*, **26** (2010), 752–759. http://dx.doi.org/10.1002/CNM.1172

20. L. Du, T. Sogabe, S. L. Zhang, A fast algorithm for solving tridiagonal quasi-Toeplitz linear systems, *Appl. Math. Lett.*, **75** (2018), 74–81. http://dx.doi.org/10.1016/j.aml.2017.06.016

21. Y. L. Wei, Y. P. Zheng, Z. L. Jiang, S. Shon, A study of determinants and inverses for periodic tridiagonal Toeplitz matrices with perturbed corners involving Mersenne numbers, *Mathematics*, **7** (2019), 893. http://dx.doi.org/10.3390/math7100893

22. Y. L. Wei, X. Y. Jiang, Z. L. Jiang, S. Shon, Determinants and inverses of perturbed periodic tridiagonal Toeplitz matrices, *Adv. Differ. Equ.*, **2019** (2019), 410. http://dx.doi.org/10.1186/s13662-019-2335-6

23. Y. L. Wei, X. Y. Jiang, Z. L. Jiang, S. Shon, On inverses and eigenpairs of periodic tridiagonal Toeplitz matrices with perturbed corners, *J. Appl. Anal. Comput.*, **10** (2020), 178–191. http://dx.doi.org/10.11948/20190105

24. T. Sogabe, Numerical algorithms for solving comrade linear systems based on tridiagonal solvers, *Appl. Math. Comput.*, **198** (2008), 117–122. http://dx.doi.org/10.1016/j.amc.2007.08.029

25. J. T. Jia, S. M. Li, New algorithms for numerically solving a class of bordered tridiagonal systems of linear equations, *Comput. Math. Appl.*, **78** (2019), 144–151. http://dx.doi.org/10.1016/j.camwa.2019.02.028

26. G. Golub, C. Van Loan, *Matrix computations*, 3 Eds., Baltimore and London: Johns Hopkins University Press, 1996.

27. C. Garoni, S. Serra-Capizzano, *Generalized locally Toeplitz sequences: Theory and applications*, Cham: Springer, 2017.
28. A. Böttcher, S. M. Grudsky, *Spectral properties of banded Toeplitz matrices*, Philadelphia: SIAM, 2005.
29. M. Bogoya, S. M. Grudsky, S. Serra-Capizzano, Fast non-Hermitian Toeplitz eigenvalue computations, joining matrixless algorithms and FDE approximation matrices, *SIAM J. Matrix Anal. Appl.*, **45** (2024), 284–305. http://doi.org/10.1137/22M1529920
30. S. Serra-Capizzano, P. Tilli, Extreme singular values and eigenvalues of non-Hermitian block Toeplitz matrices, *J. Comput. Appl. Math.*, **108** (1999), 113–130. http://dx.doi.org/10.1016/S0377-0427(99)00104-1
31. M. Batista, A. A. Karawia, The use of the Sherman-Morrison-Woodbury formula to solve cyclic block tri-diagonal and cyclic block penta-diagonal linear systems of equations, *Appl. Math. Comput.*, **210** (2009), 558–563. http://dx.doi.org/10.1016/j.amc.2009.01.003
32. R. H. Chan, X. Q. Jin, *An introduction to iterative Toeplitz solvers*, Philadelphia: SIAM, 2007. http://dx.doi.org/10.1137/1.9780898718850
33. X. Q. Jin, *Preconditioning techniques for Toeplitz systems*, Beijing: Higher Education Press, 2010.
34. M. K. Ng, *Iterative methods for Toeplitz systems*, New York: Oxford University Press, 2004.
35. S. L. Lei, Y. C. Huang, Fast algorithms for high-order numerical methods for space fractional diffusion equations, *Int. J. Comput. Math.*, **94** (2017), 1062–1078. http://dx.doi.org/10.1080/00207160.2016.1149579
36. R. H. Chan, A. M. Yip, M. K. Ng, The best circulant preconditioners for Hermitian Toeplitz systems, *SIAM J. Numer. Anal.*, **38** (2000), 876–896. http://dx.doi.org/10.1137/s0036142999354083
37. I. Gohberg, V. Olshevsky, Circulants, displacements and decompositions of matrices, *Integr. Equat. Oper. Th.*, **15** (1992), 730–743. http://dx.doi.org/10.1007/BF01200697
38. R. H. Chan, M. K. Ng, Conjugate gradient methods for Toeplitz systems, *SIAM Rev.*, **38** (1996), 427–482. http://dx.doi.org/10.1137/S0036144594276474

AIMS Press