

A FAST METHOD FOR SOLVING A BLOCK TRIDIAGONAL QUASI-TOEPLITZ LINEAR SYSTEM

SKANDER BELHAJ*, †, HCINI FAHD *, AND YULIN ZHANG‡

Abstract. This paper addresses the problem of solving block tridiagonal quasi-Toeplitz linear systems. Inspired by [9], we propose a more generalized algorithm for such systems. The algorithm is based on a block decomposition for a block tridiagonal quasi-Toeplitz matrix and the Sherman-Morrison-Woodbury inversion formula. We also compare the proposed approach to the standard block LU decomposition method. A theoretical accuracy and error analysis is also considered. All algorithms have been implemented in Matlab. Numerical experiments performed with a wide variety of test problems show the effectiveness of our algorithm in terms of efficiency, stability and robustness.

Key words. System of linear equations, Tridiagonal block quasi-Toeplitz matrix, Block LU decomposition, Sherman-Morrison-Woodbury inversion.

AMS subject classifications. 65F10,65F05,15B05,15A23.

1. Introduction. An $n \times n$ matrix A is said to be quasi Toeplitz if it is a small rank perturbation of a Toeplitz matrix.

Quasi-Toeplitz matrices arise in many mathematical and engineering investigations. For instance, the inverse of a Toeplitz matrix is quasi Toeplitz [13]. And, when we use cubic B-splines collocation method for solving nonlinear parabolic differential equations, we find that the collocation matrices are quasi-Toeplitz. Especially, when considering the Neumanns or Dirichlets type boundary conditions, the collocation matrices are tridiagonal quasi-Toeplitz [21, 16, 22, 20]. Also, in curve or surface fitting, which are encountered frequently in CAD/CAM, the collocation matrices resulted by cubic B-splines basis are also tridiagonal quasi-Toeplitz type [15]. Block quasi Toeplitz matrices arise in the numerical approximation of time-dependent partial differential equations (PDEs) by generalizations of implicit multistep formulas used in boundary value form, see [3] and references therein. The block quasi-Toeplitz matrices also appear in the solution of Quasi-Birth and Death stochastic processes [8, 6, 7]. In this case, one of the main computational problems is solving linear systems whose input matrices are block quasi Toeplitz matrices. In [9] the authors gave a fast algorithm for solving tridiagonal quasi-Toeplitz linear systems. Inspired by [9], and other than [19, 17], in this paper, we will give a fast algorithm to the block tridiagonal quasi-Toeplitz case.

2. Solving a block tridiagonal quasi-Toeplitz linear system. A block tridiagonal quasi-Toeplitz matrix is defined to be a block tridiagonal Toeplitz matrix where there are a limited number of block row changes constrained as follows: There are at most p altered block rows among the first p block rows and at most q altered block rows among the last q block rows [?], then we consider the following

*ENIT-LAMSIN, University of Tunis El Manar, BP 37, 1002, Tunis, Tunisia (skander.belhaj@lamsin.rnu.tn), (hcini-fahd@yahoo.fr)

†ISAMM, MANOUBA UNIVERSITY, CAMPUS UNIVERSITAIRE DE LA MANOUBA, 2010 TUNIS, TUNISIA

‡Centro de Matemática, Universidade do Minho, 4710-057 Braga, Portugal (zhang@math.uminho.pt).

n -by- n nonsingular block tridiagonal quasi-Toeplitz linear system

$$Nx = f \quad (2.1)$$

where N is defined by

$$N = \begin{bmatrix} A & X & & & & & \\ B^* & A & B & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & B^* & A & B \\ & & & & & Y & A \end{bmatrix}$$

with A, X, Y are matrices of size $m \times m$, B Hermitian matrix of size $m \times m$, x is the nm unknown vector and f is the nm right hand side. If $B = X = Y^*$, the matrix N becomes a symmetric block tridiagonal Toeplitz matrix [1, 2, 14, 24]. By exploiting the structure of such matrices, many algorithms have been proposed for the solving the related linear systems [4, 5, 14, 19].

This paper addresses the problem of solving block tridiagonal quasi-Toeplitz linear systems (i.e. $B \neq X$ and $B^* \neq Y$). Inspired by Du et al. [9], we propose a more general algorithm for such systems. If we define the block diagonal matrix D as

$$D = \begin{bmatrix} BX^{-1} & & & & & & \\ & I & & & & & \\ & & \ddots & & & & \\ & & & I & & & \\ & & & & & & B^*Y^{-1} \end{bmatrix},$$

then

$$\tilde{N} = DN = \begin{bmatrix} Z & B & & & & & \\ B^* & A & B & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & B^* & A & B \\ & & & & & B^* & V \end{bmatrix},$$

where $Z = BX^{-1}A$ and $V = B^*Y^{-1}A$. Therefore, the system (2.1) is equivalent to the block tridiagonal quasi-Toeplitz linear system $\tilde{N}x = \tilde{f}$, where $\tilde{f} = Df$.

In the following, we first recall the block LU factorization of the matrix \tilde{N} and then introduce an efficient algorithm for solving block tridiagonal quasi-Toeplitz linear systems, which combines proper matrix decomposition with the Sherman-Morrison-Woodbury formula.

2.1. Block LU decomposition. Assume that the matrix \tilde{N} admits the following decomposition $\tilde{N} = \tilde{L}\tilde{U}$, where

Thus, solving (2.3) is equivalent to solving the Riccati equation [18]

$$B^*\Sigma^{-1}B + \Sigma = A. \quad (2.4)$$

Equation (2.4) can be solved by a direct solver, based on the following adaptation of the algorithm presented in [18]. Then,

$$\tilde{N} = LDU + E_1M_1^T, \quad (2.5)$$

with $E_1 = [I \ 0 \ \cdots \ 0]^T$ and $M_1^T = [Z - \Sigma \ 0 \ \cdots \ 0]$.

According to Sherman-Morrison-Woodbury formula [23, 12], we obtain

$$\tilde{N}^{-1} = (LDU)^{-1} - (LDU)^{-1}E_1(I + M_1^T(LDU)^{-1}E_1)^{-1}M_1^T(LDU)^{-1}. \quad (2.6)$$

Therefore, the solution x of (2.1) is obtained as follows:

$$\begin{aligned} x &= \tilde{N}^{-1}\tilde{f} \\ &= (LDU)^{-1}\tilde{f} - (LDU)^{-1}E_1(I + M_1^T(LDU)^{-1}E_1)^{-1}M_1^T(LDU)^{-1}\tilde{f} \\ &= y - (LDU)^{-1}E_1(I + (Z - \Sigma)E_1^T(LDU)^{-1}E_1)^{-1}(Z - \Sigma)y_1, \end{aligned} \quad (2.7)$$

where y is a solution of $y = (LDU)^{-1}\tilde{f}$ and y_1 is the first block of the vector y , which can be efficiently computed through Algorithm 2.

Algorithm 2 Solving $LDUx = y$

Input: L_1, U_1, Σ and Λ are matrices of sizes $m \times m$

Output: x

1. $z_1 = y_1, z_i = y_i - L_1z_{i-1}, i = 2$ to n % solving $Lz = y$
 2. $z_n = \Lambda^{-1}z_n, y_i = \Sigma^{-1}z_i, i = 1$ to $n - 1$ % solving $Dy = z$
 3. $x_n = y_n, x_i = y_i - U_1x_{i+1}, i = n - 1$ to 1 % solving $Ux = y$
-

Finally, our algorithm for solving block tridiagonal quasi-Toeplitz linear systems can be formulated as follows:

Algorithm 3 : Algorithm for solving block tridiagonal quasi-Toeplitz linear system $Nx = f$

Input: A, B, X and Y matrix of size $m \times m$.

Output: x .

1. Compute $Z = BX^{-1}A$ and $V = B^*Y^{-1}A$.
 2. Solve (2.4) by a direct solver presented in [18].
 3. Compute L_1, U_1, Σ and Λ .
 4. Solve linear system $LDUy = f$ by Algorithm 2.
 5. Compute x by (2.7).
-

REMARK 2.2. (*Idea on the cost of Algorithm 2*) For two real matrices A and B , Algorithm 2 requires $(n + k)m^3 + \mathcal{O}(nm^2)$ flops and the storage of $8m^2 + (n + 1)m$ real numbers, where k is the number of iterations for solving the matrix equation (2.4).

2.3. Error stability. DEFINITION 2.3. *Let us denote $T = LDU$ for convenience and let A and B are the blocks of the matrix T . A quadratic matrix polynomials $\lambda^2 B^* + \lambda A + B$ is called Hermitian positive definite on the unit circle if $A = A^*$, $B = B^*$ and*

$$\varrho = \min_{\theta \in \mathbb{R}} \gamma_{\min}(B^* e^{i\theta} + A + B e^{-i\theta}) > 0$$

where $\gamma_{\min}(C)$ denotes the minimum eigenvalue of a matrix C with real eigenvalues and the eigenvalues of C are sorted in the ascending order:

$$\gamma_{\min}(C) \leq \gamma_1 \leq \gamma_2 \leq \dots$$

REMARK 2.4. *Note that the parameter ρ is tiny when the matrix polynomial $\lambda^2 B^* + \lambda A + B$ has an eigenvalue near the unit circle.*

THEOREM 2.5. *Assume that the computation of $y = T^{-1} \tilde{f}$ is backward stable in the sense that it is evaluated in the form $\tilde{y} = (T + \delta_1)^{-1} \tilde{f}$, where the perturbation δ_1 is bounded as $\|\delta_1\| = \mathcal{O}(\epsilon_{\text{machine}}) \|T\|$ (see [23]). Then according to Definition 2.3, the exact solution x of the system (2.1) satisfies*

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \mathcal{O}(\epsilon_{\text{machine}}) \|N\|^{3/2} \varrho^{-3/2} \quad (2.8)$$

where \tilde{x} is the computed solution, and $\|\cdot\|$ denotes the 2-norm.

Proof. Let $Q = I + M_1^T T^{-1} E_1$ and recall from (2.6) that

$$\tilde{N}^{-1} = T^{-1} - T^{-1} E_1 Q^{-1} M_1^T T^{-1},$$

then the exact solution is

$$x = T^{-1} \tilde{f} - T^{-1} E_1 Q^{-1} M_1^T T^{-1} \tilde{f}. \quad (2.9)$$

According to the hypothesis, we have

$$\tilde{x} = (T + \delta_1)^{-1} \tilde{f} - (T + \delta_2)^{-1} E_1 Q^{-1} M_1^T (T + \delta_1)^{-1} \tilde{f}, \quad (2.10)$$

where $\|\delta_i\| = \mathcal{O}(\epsilon_{\text{machine}}) \|T\|$, $i = 1, 2$.

Neglecting quadratic perturbation terms, we obtain

$$(T + \delta_i)^{-1} = T^{-1} - T^{-1} \delta_i T^{-1}, \quad i = 1, 2.$$

So (2.9) and (2.10) allow us to write

$$\begin{aligned} x - \tilde{x} &= T^{-1} \delta_1 T^{-1} \tilde{f} - T^{-1} E_1 Q^{-1} M_1^T T^{-1} \delta_1 T^{-1} \tilde{f} - T^{-1} \delta_2 T^{-1} E_1 Q^{-1} M_1^T T^{-1} \tilde{f} \\ &= (T^{-1} - T^{-1} E_1 Q^{-1} M_1^T T^{-1}) \delta_1 T^{-1} \tilde{f} - T^{-1} \delta_2 (T^{-1} E_1 Q^{-1} M_1^T T^{-1} \tilde{f}) \\ &= \tilde{N}^{-1} \delta_1 T^{-1} \tilde{f} + T^{-1} \delta_2 (x - T^{-1} \tilde{f}) \\ &= (\tilde{N}^{-1} \delta_1 - T^{-1} \delta_2) y + T^{-1} \delta_2 x. \end{aligned} \quad (2.11)$$

As a result, the forward error satisfies the estimate

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \|\delta_2\| \|T^{-1}\| + \left[\|\delta_1\| \|\tilde{N}^{-1}\| + \|\delta_2\| \|T^{-1}\| \right] \frac{\|y\|}{\|x\|}. \quad (2.12)$$

To derive (2.8) we observe that the perturbation matrices δ_1 and δ_2 in (2.11) possess a structure when $T^{-1}\tilde{f}$ is computed by means of the Cholesky factorization $T = RR^T$, where $R = D_1U$ and $D = D_1^T D_1$ is also a Cholesky factorization. We again assume that R is computed with high precision and contains no rounding errors. Then, omitting quadratic perturbation terms, we can write

$$\delta_i = R^T \delta_{i1} + \delta_{i2} R, \quad \|\delta_{ij}\| = \mathcal{O}(\epsilon_{machine}) \|R\|, \quad i, j = 1, 2,$$

and

$$\begin{aligned} x - \tilde{x} = & \left[\tilde{N}^{-1} R^T \delta_{11} T^{-1} + \tilde{N}^{-1} \delta_{12} R T^{-1} - T^{-1} R^T \delta_{21} T^{-1} - T^{-1} \delta_{22} R T^{-1} \right] \tilde{f} \\ & + \left[T^{-1} R^T \delta_{21} \tilde{N}^{-1} + T^{-1} \delta_{22} R \tilde{N}^{-1} \right] \tilde{f}. \end{aligned}$$

The inequality $\|T\| \leq \|\tilde{N}\| \Rightarrow \|R\tilde{N}^{-1/2}\| \leq 1$ and according to Theorem 2 in [17], we have that

$$\begin{aligned} \frac{\|x - \tilde{x}\|}{\|x\|} & \leq \mathcal{O}(\epsilon_{machine}) \|\tilde{N}\|^{3/2} \varrho^{-3/2} \\ & \leq \mathcal{O}(\epsilon_{machine}) \|(DN)\|^{3/2} \varrho^{-3/2} \\ & \leq \mathcal{O}(\epsilon_{machine}) \|N\|^{3/2} \varrho^{-3/2}, \text{ where } \|D\| = \mathcal{O}(\epsilon_{machine}). \end{aligned}$$

□

REMARK 2.6. If $\|y\| = \mathcal{O}(1)\|x\|$, then

$$\frac{\|x - \tilde{x}\|}{\|x\|} = \mathcal{O}(\epsilon_{machine}) \|N\| / \varrho.$$

REMARK 2.7. If the number of blocks is large, the computation of R with the double or higher precision is not expensive in comparison with the computation cost of $T^{-1}f$.

3. Numerical results.

4. Numerical results. In this section, we present some numerical results to illustrate the effectiveness of the proposed algorithm and we compare this algorithm with the block LU method and the Gauss algorithm [10]. The algorithms were programmed in MATLAB 9.4.0.813654 (R2018a) and the computations are done on a Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz Laptop with 8.00 GB of RAM and 1.99GHz of processor. We fixed the exact solution $x^* = [1, 1, \dots, 1]^T$. The right-hand side vector was set to be $f = Nx^*$.

Tables 4.1, 4.2, 4.3, 4.4 and 4.5 show the behaviour of the absolute accuracy Δx of the block LU method, the Gauss algorithm and our method for four different sequences of block tridiagonal quasi-Toeplitz matrices where the matrices A, B, X, Y are defined as follows :

	A	B	X	Y
Example 1	E	F	B^T	B
Example 2	E	$-F$	$I + \varepsilon * \text{ones}(3)$	$B + \varepsilon * \text{ones}(3)$
Example 3	H	K	K^T	H
Example 4	I_3	F	F	$\text{circ}(20, -8, 1)$
Example 5	M	L	L^T	L

where $\text{ones}(N)$ is an N -by- N matrix of ones and $\text{circ}(N)$ is an N -by- N circulant matrix,

$$E = \begin{bmatrix} 1.20 & -0.30 & 0.10 \\ -0.30 & 2.10 & 0.20 \\ 0.10 & 0.20 & 0.65 \end{bmatrix}, \quad F = \begin{bmatrix} 0.37 & 0.13 & 0.12 \\ -0.30 & 0.34 & 0.12 \\ 0.11 & -0.17 & 0.29 \end{bmatrix},$$

$$H = \begin{bmatrix} 0.20 & 0.20 & 0.10 \\ 0.20 & 0.15 & 0.15 \\ 0.10 & 0.15 & 0.25 \end{bmatrix},$$

$$L = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}, \quad M = \begin{bmatrix} 6.0 & 5.0 \\ 5.0 & 6.8 \end{bmatrix},$$

and $K = k_{i,j}$ is an N -by- N symmetric matrix, defined from [19] by

1. Fix a real $0 \leq \ell \leq 1/2$.
 2. For $i = 1 \dots, m$
 - For $j = 1 \dots, m$
 - $k_{i,j} = 2 * i + j$
 - end j .
- Compute $\sigma_1 = \sum_{j=1}^{i-1} k_{i,j}$, $\sigma_2 = \sum_{j=1}^m k_{i,j}$
- For $j = i \dots, m$
 - $k_{i,j} = \frac{k_{i,j}(1/2-\ell-\sigma_1)}{\sigma_2}$, $k_{j,i} = k_{i,j}$.
 - end j .
- end i .
- where E, F, G, H, L and M is taken from [11].

Table 4.1 Numerical results of Example 1

	Algorithm	$n = 2^{10}$	$n = 2^{11}$	$n = 2^{12}$	$n = 2^{13}$	$n = 2^{14}$	$n = 2^{15}$
$\ x - x^*\ $	Gauss	1.23e-12	9.40e-13	1.29e-12	1.73e-12	fails	fails
	Block LU	1.20e-12	2.02e-12	3.06e-12	4.46e-12	6.40e-12	9.12e-12
	Our algorithm	1.40e-12	6.52e-12	1.11e-11	1.69e-11	2.47e-11	3.55e-11
CPU	Gauss	0.48	2.56	18.37	100.12	fails	fails
	Block LU	0.19	0.32	1.07	3.31	11.12	38.73
	Our algorithm	0.13	0.25	0.84	1.76	6.85	20.35

Table 4.2 Numerical results of Example 2 ($\varepsilon = 4 \times 10^{-3}$)

Algorithm		$n = 2^{10}$	$n = 2^{11}$	$n = 2^{12}$	$n = 2^{13}$	$n = 2^{14}$	$n = 2^{15}$
$\ x - x^*\ $	Gauss	4.05e-12	4.40e-12	5.01e-12	6.12e-12	fails	fails
	Block <i>LU</i>	4.92e-12	6.96e-12	9.86e-12	1.39e-11	1.97e-11	2.79e-11
	Our algorithm	2.24e-12	6.84e-12	1.16e-11	1.76e-11	2.57e-11	3.70e-11
CPU	Gauss	0.58	3.37	18.68	99.9	fails	fails
	Block <i>LU</i>	0.18	0.43	1.04	3.67	12.40	45.72
	Our algorithm	0.15	0.31	0.82	2.07	8.47	21.12

Table 4.3 Numerical results of Example 3 ($\sigma = 0.4, m = 3$)

Algorithm		$n = 2^{10}$	$n = 2^{11}$	$n = 2^{12}$	$n = 2^{13}$	$n = 2^{14}$	$n = 2^{15}$
$\ x - x^*\ $	Gauss	5.87e-14	8.30e-14	1.17e-13	1.66e-13	fails	fails
	Block <i>LU</i>	3.19e-14	4.33e-14	5.99e-14	8.38e-14	1.17e-13	1.66e-13
	Our algorithm	1.53e-14	1.73e-14	2.06e-14	2.60e-14	3.43e-14	4.68e-14
CPU	Gauss	1.24	4.21	24.05	117.13	fails	fails
	Block <i>LU</i>	0.23	0.55	1.06	3.58	10.69	32.02
	Our algorithm	0.13	0.25	0.47	1.02	2.39	3.42

Table 4.4 Numerical results of Example 4

Algorithm		$n = 2^{10}$	$n = 2^{11}$	$n = 2^{12}$	$n = 2^{13}$	$n = 2^{14}$	$n = 2^{15}$
$\ x - x^*\ $	Gauss	2.44e-13	3.52e-13	6.29e-13	6.38e-13	fails	fails
	Block <i>LU</i>	3.85e-14	4.36e-14	5.22e-14	5.22e-14	8.79e-14	1.19e-13
	Our algorithm	2.38e-13	3.66e-13	5.37e-13	7.73e-13	1.10e-12	1.56e-12
CPU	Gauss	0.56	2.36	13.54	116.63	fails	fails
	Block <i>LU</i>	0.11	0.29	0.72	2.16	8.20	28.45
	Our algorithm	0.097	0.25	0.50	1.30	4.30	15.12

Table 4.5 Numerical results of Example 5

Algorithm		$n = 2^{10}$	$n = 2^{11}$	$n = 2^{12}$	$n = 2^{13}$	$n = 2^{14}$	$n = 2^{15}$
$\ x - x^*\ $	Gauss	1.74e-13	2.49e-13	3.53e-13	5.01e-13	fails	fails
	Block <i>LU</i>	1.23e-13	1.75e-13	2.48e-13	3.51e-13	4.96e-13	7.02e-13
	Our algorithm	2.63e-14	3.07e-14	3.81e-14	4.97e-14	6.72e-14	9.27e-14
CPU	Gauss	0.29	1.14	6.96	56.25	fails	fails
	Block <i>LU</i>	0.08	0.17	0.45	1.56	5.40	20.24
	Our algorithm	0.07	0.12	0.26	0.62	1.88	6.17

5. Conclusion. In this paper, we proposed a generalized method for solving systems of linear equations with block tridiagonal quasi-Toeplitz matrices. Numerical examples show that the proposed algorithm is one of the good alternatives in terms of efficiency and computational time, especially when the matrix of the linear system is very large.

Acknowledgement. The third author was partially financed by Portuguese Funds through FCT within the Project UID/MAT/00013/2013.

REFERENCES

- [1] Zhong-Zhi Bai, Gene H Golub, Lin-Zhang Lu, and Jun-Feng Yin. Block triangular and skew-hermitian splitting methods for positive-definite linear systems. *SIAM Journal on Scientific Computing*, 26(3):844–863, 2005.
- [2] Michele Benzi, Dario Bini, Daniel Kressner, Hans Munthe-Kaas, and Charles Van Loan. *Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications: Cetraro, Italy 2015*, volume 2173. Springer, 2017.
- [3] Daniele Bertaccini and Fabio Di Benedetto. Spectral analysis of nonsymmetric quasi-toeplitz matrices with applications to preconditioned multistep formulas. *SIAM J. Numer. Anal.*, 45(6):2345–2367, 2007.
- [4] Dario A Bini and Beatrice Meini. The cyclic reduction algorithm: from poisson equation to stochastic processes and beyond. *Numerical Algorithms*, 51(1):23–60, 2009.
- [5] Raymond H Chan and Michael K Ng. Conjugate gradient methods for toeplitz systems. *SIAM review*, 38(3):427–482, 1996.
- [6] B. Meini D.A. Bini, S. Massei. On functions of quasi-toeplitz matrices. *Sbornik: Mathematics*, 208(11):16–28, 2017.
- [7] B. Meini D.A. Bini, S. Massei. Semi-infinite quasi-toeplitz matrices with applications to qbd stochastic processes. *Math. Comp.*, 87:2811–2830, 2018.
- [8] B. Meini L. Robol D.A. Bini, S. Massei. On quadratic matrix equations with infinite size coefficients encountered in qbd stochastic processes. *Numer. Linear Alg. Appl*, 25(6):e2128, 2018.
- [9] Lei Du, Tomohiro Sogabe, and Shao-Liang Zhang. A fast algorithm for solving tridiagonal quasi-toeplitz linear systems. *Applied Mathematics Letters*, 75:74–81, 2018.
- [10] Giesela Engeln-Müllges and Frank Uhlig. *Numerical algorithms with C*. Springer Science & Business Media, 2013.
- [11] Chun-Hua Guo and Peter Lancaster. Iterative solution of two matrix equations. *Mathematics of Computation of the American Mathematical Society*, 68(228):1589–1603, 1999.
- [12] William W Hager. Updating the inverse of a matrix. *SIAM review*, 31(2):221–239, 1989.
- [13] T. Kailath and A. H. Sayed. Displacement structure: Theory and applications. *SIAM Rev.*, 37:297–386, 1995.
- [14] Thomas Kailath and Ali H Sayed. Displacement structure: theory and applications. *SIAM review*, 37(3):297–386, 1995.
- [15] Han Xuli Liu Chengzhi and Li Juncheng. The progressive-iterative approximation for the extension of the cubic uniform b-spline curve. *Journal of Computer-Aided Design & Computer Graphics*, accepted.
- [16] J. Saberi-Nadjafi M. Gholamian. Cubic b-splines collocation method for a class of partial integro-differential equation. *Alexandria Engineering Journal*, 57:2157–2165, 2018.
- [17] Alexander Malyshev and Miloud Sadkane. Using the sherman–morrison–woodbury inversion formula for a fast solution of tridiagonal block toeplitz systems. *Linear Algebra and its Applications*, 435(11):2693–2707, 2011.
- [18] Beatrice Meini. Matrix equations and structures: efficient solution of special discrete algebraic riccati equations. In *International Conference on Numerical Analysis and Its Applications*, pages 578–585. Springer, 2000.
- [19] Borislav V Minchev. Some algorithms for solving special tridiagonal block toeplitz linear systems. *Journal of computational and applied mathematics*, 156(1):179–200, 2003.
- [20] Jalil Rashidinia and Sanaz Jamalzadeh. Collocation method based on modified cubic b-spline for option pricing models. *Math. Commun.*, 22:89–102, 2017.
- [21] R.K. Jain R.C. Mittal. Cubic b-splines collocation method for solving nonlinear parabolic partial differential equations with neumann boundary conditions. *Commun Nonlinear Sci Numer Simulat*, 17:4616–4625, 2012.
- [22] R.K. Jain R.C. Mittal. Redefined cubic b-splines collocation method for solving convectiondiffusion equations. *Applied Mathematical Modelling*, 36:5555–5573, 2012.
- [23] Charles F Van Loan. *Matrix computations (johns hopkins studies in mathematical sciences)*, 1996.
- [24] VV Voevodin and EE Tyrtshnikov. *Computational processes with toeplitz matrices*, 1987.