Fast Rational Lanczos Method for the Toeplitz Symmetric Positive Semidefinite Matrix Functions

Lei Chen[1], Lu Zhang[2(✉)], Mengjia Wu[2], and Jianqiang Zhao[2]

[1] School of Information Engineering, Xuzhou University of Technology, Xuzhou 221018, Jiangsu, China
chenlei@xzit.edu.cn

[2] School of Mathematics and Statistics, Xuzhou University of Technology, Xuzhou 221018, Jiangsu, China

Abstract. In this paper, we use the rational Lanczos method to approximate Toeplitz matrix functions, in which the matrices are symmetric positive semidefinite (SPSD). In order to reduce the computational cost, we use the inverse of the Toeplitz matrix and the fast Fourier transform (FFT). Then, we apply this method to solve a heat equation. Numerical examples are given to show the effectiveness of the rational Lanczos method.

Keywords: Toeplitz · Matrix function · Rational lanczos method · Gohberg-Semencul formula

1 Introduction

Recently, many authors have been interested in exponential integrators which are widely used in various fields [15,16,27,32]. In the exponential integrators, one needs to compute some products of φ_i matrix functions and vectors:

$$y_i(t) = \varphi_i(-tA_m)\mathrm{v}, \quad i = 0, 1, 2, \ldots, s_1, \tag{1}$$

where A_m is an $m \times m$ matrix, s_1, t are given parameters, and v is a vector. And φ_i-functions are of the following form

$$\varphi_0(x) = \exp(x), \varphi_i(x) = \int_0^1 \frac{\exp\big((1-\xi)x\big)\xi^{i-1}}{(i-1)!} d\xi, i \in \mathbb{Z}^+. \tag{2}$$

Furthermore, the φ_i-functions satisfy the following relations

$$\varphi_i(x) = x\varphi_{i+1}(x) + \frac{1}{i}, \quad i \in \mathbb{Z}^+. \tag{3}$$

Supported by the "Peiyu" Project from Xuzhou University of Technology (Grant Number XKY2019104).

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 369, pp. 164–172, 2021.
https://doi.org/10.1007/978-3-030-72792-5_15

Toeplitz matrices have various applications [5,6]. Based on the importance of Toeplitz matrices, we want to approximate the products of the φ_i matrix functions and vectors (TMF), in which the matrices are the SPSD Toeplitz matrix. That is, in (1), the matrix A_m is the SPSD Toeplitz matrix. TMF can be applied to practical calculation problems; see [12,36] for example. Recently, some new techniques are proposed to improve network routing and performance measurement [17,39]. Based on effective user behavior and traffic analysis approaches [19,20], we can design more effective scheduling strategies to raise resources utilization [22,26] and energy-efficiency [23,24]. To test new scheduling strategies, traffic must be reconstructed in test bed [18,21,25,34,38]. Fluid model is effective model to reconstruct the bursty data traffic. In this situation, TMF can also be used to build the fluid model.

Classical methods for solving φ_i matrix functions require very high complexity [2]. Recently, Krylov subspace method has been widely studied in large-scale sparse matrix due to its high efficiency [1,3,4,7–9,29,30,40]. In this method, we only need to compute the smaller matrix functions instead of computing the large matrix functions. Moreover, rational technique could be exploited to speed up Krylov subspace method [10,11].

It is known that we can calculate Toeplitz matrix-vector products by the fast Fourier transform [5,6], and one can calculate the explicit inverse of the Toeplitz matrix by the Gohberg-Semencul formula (GS) [13,14]. These important properties can be used to accelerate the rate of convergence of the computation of TMF. In this work, we use the rational Lanczos method to compute the TMF and reduce the computational cost by using the GS.

2 Toeplitz Matrix

An $m \times m$ Toeplitz matrix T_m satisfies $(T_m)_{i,j} = t_{i-j}$ for $1 \le i, j \le m$. A circulant matrix $C_m((C_m)_{i,j} = c_{i-j})$ satisfies $c_i = c_{i-m}$, $1 \le i \le m-1$. According to [5], we know that the complexity is $\mathcal{O}(m \log m)$, if one computes the products $C_m \mathrm{u}$ and $C_m^{-1}\mathrm{u}$ for a given vector u by the FFT.

A skew-circulant matrix $S_m((S_m)_{i,j} = s_{i-j})$ satisfies $s_i = -s_{i-m}$ for $1 \le i \le m-1$. Similarly, the computational complexity of the products of $S_m \mathrm{u}$ and $S_m^{-1}\mathrm{u}$ is also $\mathcal{O}(m \log m)$ by the FFT.

In addition, by constructing a proper circulant matrix, we can compute $T_m \mathrm{u}$ in $\mathcal{O}(2m \log(2m))$ complexity by the FFT; see [5,6].

The GS for the inverse of a Toeplitz matrix T_m which is SPD is as follows [13]

$$T_m^{-1} = \frac{1}{a_1}(A_m A_m^{\intercal} - \hat{A}_m \hat{A}_m^{\intercal}), \tag{4}$$

where the matrices A_m and $\hat{A}_m$ are of the following forms

$$A_m = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_m & \ldots & a_2 & a_1 \end{bmatrix}$$

and

$$\hat{A}_m = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_m & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_2 & \ldots & a_m & 0 \end{bmatrix}.$$

Denote $\mathbf{a} = [a_1, a_2, \ldots, a_m]^\intercal$, then we can get $\mathbf{a}$ by solving the following linear system

$$T_m \mathbf{a} = \mathrm{e}_1 = [1, 0, \ldots, 0]^\intercal. \tag{5}$$

According to [31,33], by using (4), one can obtain

$$T_m^{-1}\mathrm{u} = Re(\mathrm{p}) + \hat{J} Im(\mathrm{p}) \tag{6}$$

and

$$\mathrm{p} = \frac{1}{2a_1}\left[(A_m + \hat{A}_m^\intercal)(A_m^\intercal - \hat{A}_m)\right](\mathrm{u} + \mathbf{i}\hat{J}\mathrm{u}), \tag{7}$$

where $\mathbf{i}$ is the imaginary unit and $\hat{J}$ is the anti-identity matrix, and $Re(\mathrm{p})$ is the real part of p and $Im(\mathrm{p})$ is the imaginary part of p. Thus, we can compute $T_m^{-1}\mathrm{u}$ in $\mathcal{O}(m \log m)$ operations. To construct T_m^{-1} by the GS, we need to solve the Toeplitz linear system (5). We use the PCG with Strang's preconditioner to solve (5) in this paper.

3 Rational Lanczos Method

In this section, we first introduce the Lanczos method for solving $y_i(t) = \varphi_i(-tT_m)\mathrm{v}$. By using the Lanczos algorithm for a symmetric matrix T_m, we can get a basis of a Krylov subspace

$$\mathcal{K}_n(T_m, \mathrm{v}) = \mathrm{span}\{\mathrm{v}, T_m\mathrm{v}, T_m^2\mathrm{v}, \ldots, T_m^{n-1}\mathrm{v}\}.$$

Please see [35] for the details of this algorithm.

The following formulation can be obtained by the Lanczos algorithm [35]

$$T_m U_n = U_n H_n + h_{n+1,n}\mathrm{v}_{n+1}\mathrm{e}_n^\intercal, \tag{8}$$

where $U_n = [\mathrm{u}_1, \mathrm{u}_2, \ldots, \mathrm{u}_n]$ is an $m \times n$ matrix. H_n is an $n \times n$ symmetric tridiagonal matrix, and e_n is the n-th column of the identity matrix. Therefore, we can give the following approximation

$$\varphi_i(-tT_m)\mathrm{v} \approx \hat{\beta} U_n \varphi_i(-tH_n)\mathrm{e}_1, \ \hat{\beta} = \|\mathrm{v}\|_2.$$

Therefore, the computation of large matrix functions $\varphi_i(-tT_m)$ are replaced by the computation of the small matrix functions $\varphi_i(-tH_n)$. In addition, $\varphi_i(-tH_n)$ can be effectively calculated by the function "phipade" in the software package EXPINT [2].

According to [35], we note that, for approximating $\varphi_i(-tT_m)\mathrm{v}$, the rate of convergence of the Lanczos algorithm is very slow when the 2-norm of tT_m gets larger. In order to overcome this drawback, the rational Krylov subspace method is proposed [10,11,30,40].

Let I_m be the identity matrix and $\hat{\sigma}$ is a parameter. We give the rational Lanczos algorithm as follows:

Algorithm 1: Rational Lanczos algorithm

1. Calculate $\mathrm{u}_1 = \frac{\mathrm{v}}{\|\mathrm{v}\|_2}$
2. For $i = 1, 2, \ldots, n$
3. $\quad h_{i,i} = \mathrm{u}_i^{\intercal}(I_m + \hat{\sigma}T_m)^{-1}\mathrm{u}_i$
4. $\quad \hat{\mathrm{u}}_{i+1} = (I_m + \hat{\sigma}T_m)^{-1}\mathrm{u}_i - h_{i,i}\mathrm{u}_i - h_{i-1,i}\mathrm{u}_{i-1}$
5. $\quad h_{i+1,i} = \|\hat{\mathrm{u}}_{i+1}\|_2$
6. $\quad h_{i,i+1} = h_{i+1,i}$
7. $\quad \mathrm{u}_{i+1} = \frac{\hat{\mathrm{u}}_{i+1}}{h_{i+1,i}}$
8. End

Similar to (8), we have the following formulation

$$(I_m + \hat{\sigma}T_m)^{-1}U_n = U_n H_n + h_{n+1,n}\mathrm{u}_{n+1}\mathrm{e}_n^{\intercal},\ U_n^{\intercal}U_n = I_n. \tag{9}$$

Therefore, we can approximate $\varphi_i(-tT_m)\mathrm{v}$ by

$$\varphi_i(-tT_m)\mathrm{v} \approx \hat{\beta}U_n\varphi_i(-tB_n)\mathrm{e}_1,\ \hat{\beta} = \|\mathrm{v}\|_2, \tag{10}$$

where

$$B_n = \frac{1}{\hat{\sigma}}(H_n^{-1} - I_n) + h_{n+1,n}^2\left(\frac{1}{\hat{\sigma}} + \mathrm{u}_{n+1}^{\intercal}T_m\mathrm{u}_{n+1}\right)H_n^{-1}\mathrm{e}_n\mathrm{e}_n^{\intercal}H_n^{-1} = U_n^{\intercal}T_mU_n.$$

In [11], the following error bound for approximating (10) is given.

Theorem 1. *Let $A_m = P_m^{\intercal}T_mP_m$, where P_m is the projection operator of T_m on the subspace $\mathcal{K}_n((I_m + \sigma T_m)^{-1}, \mathrm{v})$, then the approximation of $\varphi_i(-tT_m)\mathrm{v}$ on the subspace $\mathcal{K}_n((I_m + \hat{\sigma}T_m)^{-1}, \mathrm{v})$ has the following error bound*

$$\|\varphi_i(-tT_m)\mathrm{v} - \varphi_i(-t\mathrm{A}_m)\mathrm{v}\| \le \frac{D}{m^{i/2}}\|\mathrm{v}\|, \tag{11}$$

where D is a constant which depends on $\hat{\sigma}$ and i.

For the rational Lanczos algorithm, $P_m = U_nU_n^{\intercal}$, and

$$\begin{aligned}\varphi_i(-tA_m)\mathrm{v} &= \varphi_i(-tP_mT_mP_m)\mathrm{v}\\ &= U_n\varphi_i(-tU_n^{\intercal}T_mU_n)U_n^{\intercal}\mathrm{v}\\ &= \hat{\beta}U_n\varphi_i(-tB_n)\mathrm{e}_1.\end{aligned}$$

The error bound of Theorem 1 shows: Firstly, the error bound of the rational Lanczos method (11) does not depend on the 2-norm of the matrix tT_n. Secondly, if the i increases, the rate of convergence of the approximation $\varphi_i(-tT_m)\mathrm{v}$ will increase.

3.1 Implementation for the TMF Algorithm

In this section, we give the implementation of the algorithm for approximating the TMF. We note that if a Toeplitz matrix T_m is a SPSD, then $I_m + \hat{\sigma}T_m$ ($\hat{\sigma} > 0$) is a SPD Toeplitz matrix. Therefore, the GS can be used to solve the inverse of the Toeplitz matrix $I_m + \hat{\sigma}T_m$. For the computation of the TMF, the rational Lanczos algorithm using the GS is as follows:

Algorithm 2: Rational Lanczos algorithm for the TMF
1. Solve $(I_m + \hat{\sigma}T_m)\mathbf{a} = \mathrm{e}_1$
2. Run Algorithm 1, where $(I_m + \hat{\sigma}T_m)^{-1}\mathrm{u}_j$ is computed by (6) and (7)
3. Calculate $\tilde{y}_i(t) = \hat{\beta}U_n\varphi_i(-tB_n)\mathrm{e}_1$

In step 1 of Algorithm 2, the cost of solving $(I_m + \hat{\sigma}T_m)\mathbf{a} = \mathrm{e}_1$ is $\mathcal{O}(m \log m)$ [5,6]. Then, the matrix-vector products $(I_m + \hat{\sigma}T_m)^{-1}\mathrm{u}_j$ in step 2 of Algorithm 2 can be computed by using (6) and (7), and the cost of computation is $\mathcal{O}(m \log m)$. In step 3 of Algorithm 2, we need to approximate $\varphi_i(-tB_n)\mathrm{e}_1$. From [37], we know that $n \ll m$ in general. Therefore, $\varphi_i(-tB_n)\mathrm{e}_1$ can be fast approximated by the function "phipade" in the software package EXPINT [2], the computation amount is $\mathcal{O}(n^3)$. As a consequence, the computation amount of Algorithm 2 is $\mathcal{O}(nm \log m)$.

4 Numerical Examples

In this section, we show the effectiveness of the rational Lanczos algorithm to approximate $\varphi_i(-tT_m)\mathrm{v}$ by two numerical examples. In Example 1, we use MATLAB command "phipade" to calculate the exact solution $\hat{y}(t)$. In the tables of numerical examples, "m" is the size of the matrix T_m, and "$Itol$" is the accuracy of the error

$$\frac{\|\hat{y}(t) - \hat{y}_n(t)\|_2}{\|\hat{y}(t)\|_2} < Itol,$$

where $\hat{y}_n(t)$ is the approximation of $\hat{y}(t)$. "IStand" and "IRL" denote the Lanczos method and rational Lanczos method, respectively. The parameter $\hat{\sigma}$ in Algorithm 2 is $\hat{\sigma} = \frac{t}{10}$ [28].

Example 1. In the first example, we study the SPD Toeplitz matrix. The elements of the SPD Toeplitz matrix are as follows [6].

$$t_i = \frac{1}{2\pi}\int_{-\pi}^{\pi} x^4 \exp(-\mathbf{i}ix)dx, i = 0, \pm 1, \pm 2, \ldots, \pm(m-1).$$

The elements of the vector v are all 1. We approximate $\varphi_i(-tT_m)\mathrm{v}$ $(i = 1, 2, 3)$. In this example, the order of the matrix T_m is 2^{10} and the value of t changes.

It can be seen from Tables 1, 2 and 3 that the numbers of iterations of the IRL are much less than these of the IStand, especially when the 2-norm of tT_m gets larger. In addition, for the IRL, the numbers of iterations do not change. This indicates that the rate of convergence of the IRL does not depend on the 2-norm of tT_m compared with the IStand.

Table 1. Numerical results for Example 1 ($i = 1$)

		$Itol = 10^{-4}$		$Itol = 10^{-7}$	
$m = 2^{10}$	t	IStand	IRL	IStand	IRL
	1	17	6	30	10
	10	56	6	91	11
	10^2	177	6	286	13
	10^3	562	6	880	13

Table 2. Numerical results for Example 1 ($i = 2$)

		$Itol = 10^{-4}$		$Itol = 10^{-7}$	
$m = 2^{10}$	t	IStand	IRL	IStand	IRL
	1	15	5	27	9
	10	48	5	83	10
	10^2	155	5	261	10
	10^3	493	6	807	12

To compare the computational time of the IStand and the IRL, we give the results of the numbers of iterations and computational time in seconds of the IStand and the IRL in Table 4, where $Itol = 10^{-9}$ and $m = 2^{10}$. It can be seen from Table 4: Firstly, the computational times and the numbers of iterations of the IRL are much less than these of the IStand. Furthermore, if the size of the matrix T_m gets larger, the superiority of the IRL will become more obvious. Secondly, if t is fixed, as i increases, the iteration numbers of the IRL decreases, which also validates the result of (11) in Theorem 1.

Example 2. In the second example, we study a heat equation [12]. Please refer to [12] for the detailed equation. Numerically solving the heat equation leads to a matrix function problem

$$\hat{v}(t) = (-tT_m)\varphi_1(-tT_m)v_0 + v_0,$$

Table 3. Numerical results for Example 1 ($i = 3$)

		$Itol = 10^{-4}$		$Itol = 10^{-7}$	
$m = 2^{10}$	t	IStand	IRL	IStand	IRL
	1	13	4	25	9
	10	43	5	77	9
	10^2	139	5	242	10
	10^3	444	5	752	10

Table 4. Numerical results of the IRL and the IStand for Example 1

	$i=1$		$i=2$		$i=3$	
t	IStand	IRL	IStand	IRL	IStand	IRL
1	36(0.0107)	14(0.0030)	33(0.0113)	13(0.0042)	31(0.0073)	12(0.0014)
10	110(0.0261)	17(0.0041)	102(0.0246)	14(0.0012)	95(0.1597)	13(0.0056)
10^2	343(0.0524)	17(0.0043)	319(2.6682)	14(0.0019)	300(1.5130)	14(0.0063)
10^3	1050(59.1512)	17(0.0050)	980(77.9534)	16(0.1218)	915(85.0608)	14(0.0071)

where

$$\hat{v}(t) = [\hat{v}_1(t), \hat{v}_2(t), \ldots, \hat{v}_m(t)]^\intercal$$

is an approximation solution, T_m is a SPD Toeplitz matrix, and v_0 is an initial vector. We solve $\hat{v}(t)$ by the IStand and the IRL, respectively. Table 5 lists the numbers of iterations and computational times of the IStand and the IRL for different m and t.

Table 5. Numerical results for Example 2

	$t=60$			$t=300$		
m	Ier	IRL	IStand	Ier	IRL	IStand
2^7	7.88×10^{-5}	12(0.0020)	60(0.0192)	6.71×10^{-5}	12(0.0011)	64(0.0231)
2^8	1.97×10^{-5}	14(0.0022)	142(0.0753)	1.68×10^{-5}	12(0.0011)	128(0.1570)
2^9	4.92×10^{-6}	14(0.0076)	270(0.9505)	4.19×10^{-6}	12(0.0069)	256(1.0489)
2^{10}	1.23×10^{-6}	14(0.0080)	510(8.0049)	1.05×10^{-6}	12(0.0241)	512(9.4987)
2^{11}	3.08×10^{-7}	14(0.0110)	1020(56.4147)	2.62×10^{-7}	12(0.0739)	1024(65.0765)
2^{12}	7.70×10^{-8}	14(0.0231)	2039(384.2013)	6.55×10^{-8}	12(0.2682)	2048(450.0337)
2^{13}	1.92×10^{-8}	14(0.0501)	> 3600	1.66×10^{-8}	12(0.3201)	> 3600

According to Table 5, it is seen that the IRL needs fewer numbers of iterations and calculation times to reach the final accuracies than these of the IStand. In addition, for the large matrix size, the IStand becomes unacceptable due to a lot of iteration numbers, while the IRL still works well.

5 Conclusion and Future Work

In this work, we use the rational Lanczos algorithm to approximate the TMF, and this method is applied to the numerical calculation. Using the GS, we can avoid the use of internal iterations to implement the rational Lanczos algorithm. In addition, due to the Toeplitz matrix, the amount of computation can be reduced. Numerical results show the advantage of the new method.

References

1. Al-Mohy, A., Higham, N.J.: Computing the action of matrix exponential, with an application to exponential integrators. SIAM J. Sci. Comput. **33**, 488–511 (2011)
2. Berland, H., Skaflestad, B., Wright, W.: Expint-A matlab package for exponential integrators. ACM Tran. Math. Soft. **33**(4), 4-es (2007)
3. Botchev, M., Grimm, V., Hochbruck, M.: Residual, restarting and Richardson iteration for the matrix exponential. SIAM J. Sci. Comput. **35**, A1376–A1397 (2013)
4. Caliari, M., Kandolf, P., Zivcovich, F.: Backward error analysis of polynomial approximations for computing the action of the matrix exponential. BIT Numer. Math. **58**, 907–935 (2018)
5. Chan, R., Jin, X.: An Introduction to Iterative Toeplitz Solvers. SIAM, Philadelphia (2007)
6. Chan, R., Ng, M.: Conjugate gradient methods for Toeplitz systems. SIAM Rev. **38**, 427–482 (1996)
7. Eiermann, M., Ernst, O.: A restarted Krylov subspace method for the evaluation of matrix functions. SIAM J. Numer. Anal. **44**, 2481–2504 (2006)
8. Frommer, A., Gttel, S., Schweitzer, M.: Efficient and stable Arnoldi restarts for matrix functions based on quadrature. SIAM J. Matrix Anal. Appl. **35**, 661–683 (2014)
9. Frommer, A., Simoncini, V.: Stopping criteria for rational matrix functions of hermitian and symmetric matrices. SIAM J. Sci. Comput. **30**, 1387–1412 (2008)
10. Gčkler, T., Grimm, V.: Uniform approximation of φ functions in exponential integrators by a rational Krylov subspace method with simple poles. SIAM J. Matrix Anal. Appl. **35**, 1467–1489 (2014)
11. Grimm, V.: Resolvent Krylov subspace approximation to operator functions. BIT Numer. Math. **52**, 639–659 (2012)
12. Gockenbach, M.: Partial Differential Equations-Analytical and Numerical Methods. SIAM, Philadelphia (2002)
13. Gohberg, I., Semencul, A.: On the inversion of finite Toeplitz matrices and their continuous analogs. Matem. Issled. **2**, 201–233 (1972)
14. Heinig, G., Rost, L.: Algebraic Methods for Toeplitz-like Matrices and Operators. Birkhäuser, Basel (1984). https://doi.org/10.1007/978-3-0348-6241-7
15. Higham, N.J., Kandolf, P.: Computing the action of trigonometric and hyperbolic matrix functions. SIAM J. Sci. Comput. **36**, A613–A627 (2017)
16. Hochbruck, M., Ostermann, A.: Exponential integrators. Acta Numer. **19**, 209–286 (2010)
17. Huo, L., Jiang, D., Lv, Z., et al.: An intelligent optimization-based traffic information acquirement approach to software-defined networking. Comput. Intell. **36**, 1–21 (2019)
18. Huo, L., Jiang, D., Qi, S., Song, H., Miao, L.: An AI-based adaptive cognitive modeling and measurement method of network traffic for EIS. Mob. Net. Appl., 1–11 (2019). https://doi.org/10.1007/s11036-019-01419-z
19. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Netw. Sci. Eng. **1**(1), 1–12 (2018)
20. Jiang, D., Wang, Y., Lv, Z., et al.: Big data analysis-based network behavior insight of cellular networks for industry 4.0 applications. IEEE Trans. Ind. Inf. **16**(2), 1310–1320 (2020)

21. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS One **13**(5), 1–23 (2018)
22. Jiang, D., Huo, L., Lv, Z., et al.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **19**(10), 3305–3319 (2018)
23. Jiang, D., Li, W., Lv, H.: An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. Neurocomputing **220**, 160–169 (2017)
24. Jiang, D., Wang, Y., Lv, Z., et al.: Intelligent Optimization-based reliable energy-efficient networking in cloud services for IIoT networks. IEEE J. Sel. Areas Commun. Online available (2019)
25. Jiang, D., Wang, W., Shi, L., et al.: A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans. Netw. Sci. Eng. **5**(3), 1–12 (2018)
26. Jiang, D., Zhang, P., Lv, Z., et al.: Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. IEEE Internet Things J. **3**(6), 1437–1447 (2016)
27. Kooij, G.L., Botchev, M.A., Geurts, B.J.: An exponential time integrator for the incompressible Navier-stokes equation. SIAM J. Sci. Comput. **40**, B684–B705 (2018)
28. Lee, S., Pang, H., Sun, H.: Shift-invert Arnoldi approximation to the Toeplitz matrix exponential. SIAM J. Sci. Comput. **32**, 774–792 (2010)
29. Lopez, L., Simoncini, V.: Analysis of projection methods for rational function approximation to the matrix exponential. SIAM J. Numer. Anal. **44**, 613–635 (2006)
30. Moret, I.: On RD-rational Krylov approximations to the core-functions of exponential integrators. Numer. Linear Algebra Appl. **14**, 445–457 (2007)
31. Ng, M., Sun, H., Jin, X.: Recursive-based PCG methods for Toeplitz systems with nonnegative generating functions. SIAM J. Sci. Comput. **24**, 1507–1529 (2003)
32. Noferini, V.: A formula for the Fréchet derivative of a generalized matrix function. SIAM J. Matrix Anal. Appl. **38**, 434–457 (2017)
33. Pang, H., Sun, H.: Shift-invert Lanczos method for the symmetric positive semidefinite Toeplitz matrix exponential. Numer. Linear Algebra Appl. **18**, 603–614 (2011)
34. Qi S., Jiang, D., Huo, L.: A prediction approach to end-to-end traffic in space information networks. Mob. Netw. Appl. Online available (2019)
35. Saad, Y.: Analysis of some Krylov subspace approximations to the matrix exponential operator. SIAM J. Numer. Anal. **29**, 209–228 (1992)
36. Tangman, D.Y., Gopaul, A., Bhuruth, M.: Exponential time integration and Chebychev discretisation schemes for fast pricing of options. Appl. Numer. Math. **58**, 1309–1319 (2008)
37. Van Den Eshof, J., Hochbruck, M.: Preconditioning Lanczos approximations to the matrix exponential. SIAM J. Sci. Comput. **27**, 1438–1457 (2006)
38. Wang, Y., Jiang, D., Huo, L., et al.: A new traffic prediction algorithm to software defined networking. Mob. Netw. Appl. Online available (2019)
39. Wang, F., Jiang, D., Qi, S.: An adaptive routing algorithm for integrated information networks. China Commun. **7**(1), 196–207 (2019)
40. Wu, G., Feng, T., Wei, Y.: An inexact shift-and-invert Arnoldi algorithm for Toeplitz matrix exponential. Numer. Linear Algebra Appl. **22**, 777–792 (2015)