Shift-invert rational Krylov method for evolution equations

Yuka Hashimoto[1] Takashi Nodera[2]

Received 19 December 2016; revised 30 November 2017

Abstract

In order to obtain the numerical solution of evolution equations which arise in various fields of science and technology, the computation of matrix functions called ϕ-functions is required. This paper proposes a new method called the shift-invert rational Krylov method for the computation of matrix ϕ-functions. This method efficiently computes the matrix ϕ-functions and allows the appropriate parameters to be simply determined.

Subject class: 65F60, 65M22

Keywords: Shift-invert rational Krylov, ϕ-function

DOI:10.21914/anziamj.v58i0.11622, © Austral. Mathematical Soc. 2017. Published 2017-12-04, as part of the Proceedings of the 18th Biennial Computational Techniques and Applications Conference. ISSN 1445-8810. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to the DOI for this article. Record comments on this article via
http://journal.austms.org.au/ojs/index.php/ANZIAMJ/comment/add/11622/0

Contents

1 Introduction

Evolution equations are utilized in various fields of science and technology. Primary examples are the heat equation and the wave equation, both of which occur in fluid dynamics. Let $\Omega \subseteq \mathbb{R}^d$ be an open set, $T > 0$, $l \in \mathbb{N}$, and $\mathcal{V} \subseteq L^2(\Omega)$ be a Hilbert space. Assume $\mathcal{D}$ is a differential operator on $\mathcal{V}$. Then, consider the function $u \in C^l((0,T]) \times \mathcal{V}$ which satisfies

$$\frac{\partial^l u}{\partial t^l} = \mathcal{D}u, \tag{1}$$

with some appropriate initial and boundary conditions. When equation (1) is discretized in space using a finite element method, a differential-algebraic equation of the following form results

$$M\dot{y}(t) = F(y(t)), \quad y(0) = v,$$

where $M \in \mathbb{R}^{n\times n}$ is an invertible matrix, and F is a vector valued function. To solve this system, the exponential integrators are an effective method

to undertake the time integration [5, 6]. At the ith step, an exponential integrator rearranges F as $F(y) = L_i y + n_i(y)$, and computes the scheme

$$\begin{cases} Y_{ir} = \phi_0(c_r \Delta t M^{-1} L_i) y_i + \Delta t \sum_{l=1}^{r-1} a_{rl}(\Delta t M^{-1} L_i) M^{-1} n_i(Y_{il}), \\ y_{i+1} = \phi_0(\Delta t M^{-1} L_i) y_i + \Delta t \sum_{r=1}^{s} b_r(\Delta t M^{-1} L_i) M^{-1} n_i(Y_{ik}), \end{cases} \tag{2}$$

where Δt is the temporal step size, and a_{rl}, b_r are the linear combinations of ϕ_k $(k \leqslant r)$, which are known as ϕ-functions. The ϕ-functions are defined as

$$\begin{aligned} &\phi_0(z) := e^z, \\ &\phi_k(z) := \frac{\phi_{k-1}(z) - \frac{1}{(k-1)!}}{z}, \quad k = 1, 2, \ldots. \end{aligned}$$

The simplest scheme is

$$y_{i+1} = e^{\Delta t M^{-1} L_i} y_i + \Delta t \phi_1(\Delta t M^{-1} L_i) M^{-1} n_i(y_i).$$

There are various methods for computing matrix ϕ-functions. Krylov subspace methods are a valid option, since it is sufficient for scheme (2) to compute the product $\phi_k(A)v$, where $A = \Delta t M^{-1} L_i$ and $v = M^{-1} n_i(Y_{ik})$. Krylov subspace methods approximate the matrix function multiplied with a vector on a subspace of dimension smaller than n. The most simple and well-known method is the Arnoldi method for ϕ-functions (AP), but the convergence of AP depends on the width of the numerical range of A [5]. To address this issue, the rational Krylov method for ϕ-functions (RKP) was proposed by Beckermann and Reichel [1]. Göckler [3] showed that the RKP converges independently of the width of the numerical range of A. However, it requires the calculation of parameters called shifts during every step of the Krylov process. Methods for choosing the optimal shifts for symmetric matrices have been proposed by Güttel [4]. Göckler [3] has proposed methods for choosing shifts for nonsymmetric matrices. However, since the shifts are complex, even

if the matrices M and L_i are real, complex values appear due to the shifts, which increases computational costs. To solve this problem, a new method called the shift-invert rational Krylov method (SIRKP) is investigated in this paper. In this method, the appropriate real-valued shifts are determined, which realize a faster convergence.

2 Rational Krylov method

In the following sections, the algorithm for the calculation of $\phi_k(A)v$ is outlined. We define the numerical range of A as $W(A) := \{u^*Au \mid u \in \mathbb{C}^n,\ \|u\| = 1\}$, and assume $W(A) \subseteq \mathbb{C}^-$, where $\mathbb{C}^\pm := \{z \in \mathbb{C} \mid \Re(z) \gtrless 0\}$, and $\|\cdot\| = \|\cdot\|_2$.

The m-step rational Krylov process with the initial vector $v_1 = v/\|v\|$ is

$$h_{j+1,j}v_{j+1} = (\gamma_j I - A)^{-1}v_j - \sum_{k=1}^{j} h_{k,j}v_k, \tag{3}$$

where $h_{k,j} = v_k^*(\gamma_j I - A)^{-1}v_j$, $h_{j+1,j} = \|(\gamma_j I - A)^{-1}v_j - \sum_{k=1}^{j} h_{k,j}v_k\|$, and $\gamma_j > 0$ is a different shift in every step for $j = 1, \ldots, m$. This results in the matrix relation

$$V_m = V_m H_m D_m - AV_m H_m + (\gamma_m I - A)h_{m+1,m}v_{m+1}e_m^*,$$

where $V_m = [v_1, \ldots, v_m]$, H_m is an upper Hessenberg matrix, $D_m := \operatorname{diag}\{\gamma_1, \ldots, \gamma_m\}$, and e_j is the jth column of the identity matrix. It follows that

$$\phi_k(A)v \approx V_m\phi_k(V_m^*AV_m)V_m^*v = r(A)v, \tag{4}$$

for $r \in \mathcal{P}_m/q_m$, where $q_m(x) = (\gamma_1 - x)\cdots(\gamma_m - x)$, $\mathcal{P}_m$ represents the polynomials of degree less than or equal to m, and $\mathcal{P}_m/q_m := \{p_m/q_m;\ p_m \in \mathcal{P}_m\}$.

3 Shift-invert rational Krylov method

The m-step rational Krylov process is computed using (3). However, this study uses the shifts $\gamma_j = N - j \in \mathbb{R}$, where $N \in \mathbb{N}$ satisfies $\gamma_j > 0$ for all $j = 1, \dots, m$. This results in the relation

$$V_m^*(\gamma_m I - A)^{-1} V_m = H_m(I - H_m D_m + \gamma_m H_m)^{-1} =: K_m, \tag{5}$$

where $D_m := \operatorname{diag}\{\gamma_1, \dots, \gamma_m\}$. Then, $\phi_k(A)$ is regarded as a function of $(\gamma_m I - A)^{-1}$, $f_m((\gamma_m I - A)^{-1})v$, where $f_m(x) := \phi_k\left(\gamma_m - x^{-1}\right)$, and

$$\begin{aligned} \phi_k(A)v &= f_m((\gamma_m I - A)^{-1})v \\ &\approx V_m f_m(V_m^*(\gamma_m I - A)^{-1} V_m) V_m^* v \\ &= V_m f_m(K_m) V_m^* v. \end{aligned} \tag{6}$$

In the RKP, the approximation (4) uses the same function in each step. On the other hand, the approximation (6) depends on m, and changes at every step. As do the matrices projected on to the rational Krylov subspace.

Remark 1. The approximation (6) can be transformed to

$$V_m f_m(K_m) V_m^* v = V_m \phi_k\left((H_m D_m - I) H_m^{-1}\right) V_m^* v. \tag{7}$$

This is used in the computations here. In SIRKP, $(I - H_m D_m) H_m^{-1}$ is used instead of $V_m^* A V_m$, which appears in the approximation (4). The matrices H_m and D_m are available with no additional cost, and since $m \ll n$, the computation of approximation (7) is numerically efficient.

In order to analyze the convergence of approximation (6), the space constructed by SIRKP is investigated. Let $X_j := (\gamma_j I - A)^{-1}$ for $j = 1, \dots, m$. Given $\gamma_j = N - j$, thus

$$X_j = (\gamma_j I - A)^{-1} = (I - (\gamma_m - \gamma_j) X_m)^{-1} X_m = (I + (m - j) X_m)^{-1} X_m.$$

The space generated by the m-step SIRKP is therefore

$$\begin{aligned}&\operatorname{Span}\{v,\ X_1 v,\ \ldots,\ X_m v\}\\&= \operatorname{Span}\{v,\ (I+(m-1)X_m)^{-1}X_m v,\ \ldots,\ (I+X_m)^{-1}X_m v,\ X_m v\}\\&= \{r(X_m)v \mid\ r \in \mathcal{P}_{m-1}/q_{m-1},\ q_m(x) = (1+mx)\cdots(1+x)\}.\end{aligned}$$

The space generated up to the mth step of SIRKP is the rational Krylov subspace generated by matrix X_m. The following properties regarding the approximation of SIRKP are derived from these observations and the result of Beckermann and Reichel [1].

Proposition 2. *Let $q_m(x) := (1+mx)\cdots(1+x)$ and $\mathcal{P}_m$ be the set of polynomials of degree less than or equal to m. For all $r \in \mathcal{P}_{m-1}/q_{m-1}$, then*

$$r(X_m)v = V_m r(K_m) V_m^* v. \tag{8}$$

Proposition 3. *There exists $r_m \in \mathcal{P}_{m-1}/q_{m-1}$ such that*

$$V_m f_m(K_m) V_m^* v = r_m(X_m)v. \tag{9}$$

Remark 4. The consequence of Proposition 2 is that the approximation of the rational functions in $\mathcal{P}_{m-1}/q_{m-1}$ of X_m with SIRKP is exact. Moreover, Proposition 3 implies that all the approximations of SIRKP are represented by some rational function in $\mathcal{P}_{m-1}/q_{m-1}$ of X_m.

We therefore obtain the following theorem regarding the convergence of SIRKP.

Theorem 5. *Let $\mathcal{H}(\Pi)$ be the set of holomorphic functions in the closed and bounded set: $\Pi \subseteq \mathbb{C}$ to $\mathbb{C}$. Let $1 \leqslant C \leqslant 11.08$, and $f(x) := \int_0^1 e^{N-sx^{-1}}(1-s)^{k-1}/(k-1)!\,ds$ for $k \geqslant 1$, $f(x) = e^{N-x^{-1}}$ for $k = 0$. It is possible to choose the closed and bounded set Σ containing $\cup_{j=1}^{N-1} W(X_j)$ and satisfying $\Sigma \subseteq \mathbb{C}^+$. With the Σ, for $m = 1, \ldots, N-1$, the error bound of SIRKP is estimated as*

$$\|\phi_k(A)v - V_m f_m(K_m) V_m^* v\| \leqslant 2C\|v\| e^{-m} \times \min_{r \in \mathcal{P}_{m-1}/q_{m-1}} \|f - r\|_\Sigma, \tag{10}$$

where $\|\cdot\|_\Sigma$ is the norm in $\mathcal{H}(\Sigma)$, which is defined as $\|g\|_\Sigma = \sup_{x\in\Sigma} |g(x)|$.

Proof: Since $W(A) \subseteq \mathbb{C}^-$ and $\gamma_j = N - j > 0$, $W(X_j) \subseteq \mathbb{C}^+$ for all $j = 1, \ldots, N-1$. In addition, $W(X_j)$ are bounded. This makes it possible to choose a closed and bounded set $\Sigma \subseteq \mathbb{C}^+$ which contains $\cup_{j=1}^{N-1} W(X_j)$.

Let $r \in \mathcal{Q}_m$ be arbitrary. Then $\phi_k(A) = f_m(X_m)$, and Proposition 2 yields

$$\begin{aligned}\|\phi_k(A)v - V_m f_m(K_m)V_m^* v\| \\ = \|f_m(X_m)v - r(X_m)v - V_m f_m(K_m)V_m^* v + V_m r(K_m)V_m^* v\|. \end{aligned} \quad (11)$$

Since all the poles of functions in $\mathcal{P}_{m-1}/q_{m-1}$ are real and negative, then $\mathcal{P}_{m-1}/q_{m-1} \subseteq \mathcal{H}(\Sigma)$, and $f_m,\ f \in \mathcal{H}(\Sigma)$. Moreover, from equation (5), $W(K_m) \subseteq W(X_m)$. Following Crouzeix [2], there is $C \in [1, 11.08]$ such that

$$\|f_m(X_m) - r(X_m)\| \leqslant C\|f_m - r\|_\Sigma, \quad \|f_m(K_m) - r(K_m)\| \leqslant C\|f_m - r\|_\Sigma. \quad (12)$$

Let $\mathcal{Q}_m = \mathcal{P}_{m-1}/q_{m-1}$. Since ϕ_k is represented as $\phi_k(x) = \int_0^1 e^{sx}(1-s)^{k-1}/(k-1)!\, ds$ for $k \geqslant 1$, and $\gamma_j = N - j$, it is deduced using (11) and (12) that

$$\begin{aligned}
&\|\phi_k(A)v - V_m f_m(K_m)V_m^* v\| \\
&\quad \leqslant 2C\|v\| \min_{r\in\mathcal{Q}_m} \|f_m - r\|_\Sigma \\
&\quad = 2C\|v\| \min_{r\in\mathcal{Q}_m} \sup_{z\in\Sigma} |\phi_k(N - m - z^{-1}) - r(z)| \\
&\quad = 2C\|v\| \min_{r\in\mathcal{Q}_m} \sup_{z\in\Sigma} \left| \int_0^1 e^{s(N-m-z^{-1})} \frac{(1-s)^{k-1}}{(k-1)!} - e^{s(N-m)} e^{-s(N-m)} r(z)\, ds \right| \\
&\quad \leqslant 2C\|v\| \min_{r\in\mathcal{Q}_m} \sup_{z\in\Sigma} e^{-m} \left| \int_0^1 e^{N-sz^{-1}} \frac{(1-s)^{k-1}}{(k-1)!}\, ds - \int_0^1 e^{N-s(N-m)}\, ds\, r(z) \right| \\
&\quad = 2C\|v\| \min_{r\in\mathcal{Q}_m} e^{-m} \sup_{z\in\Sigma} \left| \int_0^1 e^{N-sz^{-1}} \frac{(1-s)^{k-1}}{(k-1)!}\, ds - r(z) \right|,
\end{aligned}$$

for $k \geqslant 1$. The proof for $k = 0$ is similar. ♠

In the error bound (10), the term e^{-m} becomes smaller as m becomes larger. In addition, choosing γ_j as $N - j$ makes the space $\mathcal{P}_{m-1}/q_{m-1}$ expand with

each iteration, since q_m has the form $q_m(x) = (1+mx)\cdots(1+x)$. Therefore, we also have that $\min_{r\in\mathcal{P}_{m-1}/q_{m-1}} \|f-r\|_\Sigma$ becomes smaller as m becomes larger. A comparison of this to the upper bound of RKP [3, Theorem 4.16], illustrates that the term e^{-m} accelerates the convergence of SIRKP.

Remark 6. It is uncertain in advance whether or not $N > j$ for all j. However, if j reaches N in the middle of the algorithm, it can be reset with a new larger N. In equation (10), the function f is dependent on N. Thus, the error may not decrease when N changes, but it will decrease from this point. Moreover, when setting the maximum iteration number to $m^{\max}$, setting $N = m^{\max}+1$ ensures $N > j$ for all j. In this case, the shifts of SIRKP are completely determined.

4 Numerical experiments

All numerical computations in this section were done with C on an Intel(R) Xeon(R) X5690 3.47GHz processor with the Ubuntu14.04LTS operating system.

Example 7. Consider the wave equation on the rectangular region $\Omega = (0,1)\times(0,1) \subseteq \mathbb{R}^2$:

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - c^2\Delta u = f(x_1,x_2,t) \quad \text{in } (0,T]\times\Omega, \\ f(x_1,x_2,t) = -10^4 \sin(t) e^{(x_1-0.8)^2+(x_2-0.8)^2}, \\ u = e^{-10(x_1-0.5)^2-10(x_2-0.5)^2} \quad \text{on } \{0\}\times\Omega, \\ \dfrac{\partial u}{\partial t} = 0 \quad \text{on } \{0\}\times\Omega, \\ u = 0 \quad \text{on } (0,T]\times\partial\Omega_1, \qquad \dfrac{\partial u}{\partial n} = 0 \quad \text{on } (0,T]\times\partial\Omega_2, \end{cases} \tag{13}$$

where $\partial\Omega_1 = [0,1]\times\{0\}\bigcup[0,1]\times\{1\}$, $\partial\Omega_2 = \partial\Omega \setminus \partial\Omega_1$, and $c = \sqrt{0.1}$. In order to confirm the effectiveness of the exponential integrator (EI), this

was compared with the equivalent results for the implicit Euler method (IE). Choosing $n = 1312$, the solution was computed to $t = 50$. The solutions are shown in Figure 1. Observe that IE with $\Delta t = 0.01$ dampens the solution. The CPU time of EI with $\Delta t = 0.1$ was 9.4 s, and that of IE with $\Delta t = 10^{-3}$ was 34.0 s. Hence, the converged solution was computed by EI around four times faster than by IE. In the next step, the effectiveness of SIRKP in computing matrix ϕ-functions was tested. A larger region and a finer mesh were set with $\Omega = (-1.5, 1.5) \times (-1, 1)$, and $n = 237378$. The matrix ϕ_1 function and the vector product which appeared in the exponential integrator were computed up to a relative error tolerance of 10^{-6}. The CPU time and the number of iterations for computing the vector product to accuracy 10^{-6} with the shift-invert Arnoldi method (SIAP) [3], RKP, and SIRKP are shown in Table 1. The relative error of each algorithm as a function of iteration number are shown in Figure 2. The shifts $\gamma_j = r + h \cdot (-1)^{j-1} \lceil (j-1)/2 \rceil i$ introduced by Göckler [3] for RKP, were employed. In addition, we set $m^{\max} = 50$ for SIRKP, and $\gamma_j/\Delta t$ was used instead of γ_j in order to treat $M^{-1}L_{i+1}$ instead of $\Delta t M^{-1}L_{i+1}$. SIAP caused numerical instability or converged very slowly if the shift was not appropriate. RKP used complex shifts even though the matrices M, L_{i+1} and vector v were real, which required additional computational cost. Moreover, it failed to converge, due to instability in the computation of $V_m^* A V_m$ in (4). Alternative values of r and h were used, but the results did not change. On the other hand, SIRKP converged at a similar rate to the optimal value for SIAP. Since the shifts of SIRKP changes at every step, the impact of each shift is balanced. This effect of SIRKP is explained due to the term e^{-m} in equation (10).

5 Conclusion

Exponential integrators, which are used for the numerical solution of algebraic-differential equations, require the efficient calculation of matrix ϕ-functions. The SIRKP method has been shown in this paper to be an efficient method to

Figure 1: Numerical solutions of the problem (13) in Example 7, which is computed to $t = 50$ (a): with EI of $\Delta t = 0.1$ (b): with IE of $\Delta t = 0.01$ (c): with IE of $\Delta t = 10^{-3}$).

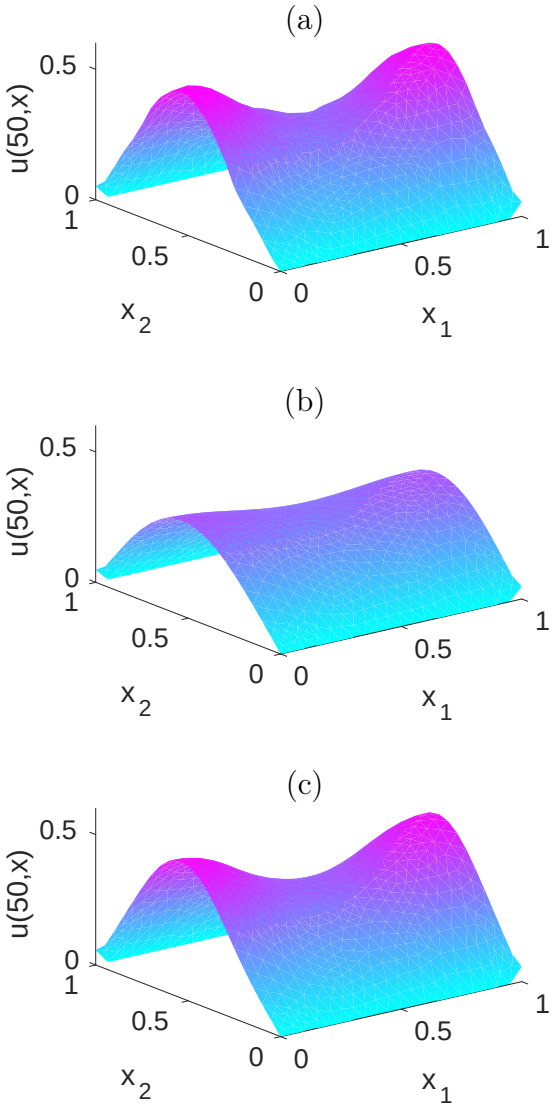
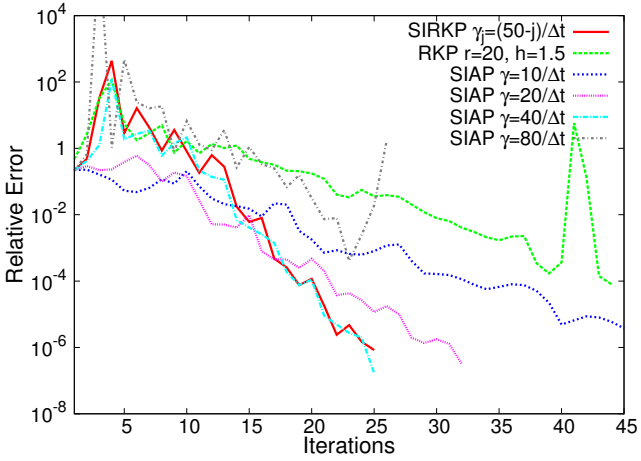


Table 1: CPU time and iterations of SIAP, RKP and SIRKP for the computation of the matrix ϕ_1-function and the vector product in Example 7 up to a relative error tolerance of 10^{-6}. (RKP with $r = 20/\Delta t$, $h = 1.5/\Delta t$ and SIAP with $\gamma = 80/\Delta t$ caused numerical instability before the relative errors reached 10^{-6}.)

Algorithm	γ_j	CPU time (s)	Iterations
SIRKP	$(50-j)/\Delta t$	7.1	25
SIAP	$10/\Delta t$	51	51
SIAP	$20/\Delta t$	14	32
SIAP	$40/\Delta t$	6.4	25

Figure 2: Number of iterations versus relative errors for SIAP, RKP, and SIRKP for the computation of the matrix ϕ_1-function and the vector product in Example 7.



calculate ϕ-functions which determines the necessary real-valued shifts. The main difference between this and the existing method of RKP, is its choice of shifts and the way the approximation is made.

References

[1] B. Beckermann and L. Reichel, Error estimation and evaluation of matrix functions via the Faber transform. *SIAM Journal on Numerical Analysis* **47(5)**:3849–3883, 2009. doi:10.1137/080741744 C151, C154

[2] M. Crouzeix, Numerical range and functional calculus in Hilbert space. *Journal of Functional Analysis* **244**:668–690, 2007. doi:10.1016/j.jfa.2006.10.013 C155

[3] T. Göckler, Rational Krylov subspace methods for ϕ-functions in exponential integrators. *Karlsruher Instituts für Technologie*, 2014, Ph.D. thesis. http://d-nb.info/1060425408/34 C151, C156, C157

[4] S. Güttel, Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM-Mitteilungen* **38(1)**:8–31, 2013. doi:10.1002/gamm.201310002 C151

[5] Y. Hashimoto and T. Nodera, Inexact shift-invert Arnoldi method for evolution equations. *ANZIAM Journal* **58(E)**:E1–E27, 2016. doi:10.21914/anziamj.v58i0.10766 C151

[6] M. Hochbruck and A. Ostermann, Exponential integrators. *Acta Numerica* **19**:209–286, 2010. doi:10.1017/S0962492910000048 C151

Author addresses

1. **Yuka Hashimoto**, School of Fundamental Science and Technology, Graduate School of Science and Technology, Keio University, 3-14-1

Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522, Japan and Center for Advanced Intelligence Project, RIKEN, 1-4-1 Nihonbashi, Chuo, Tokyo, 103-0027, Japan.
mailto:yukahashimoto@math.keio.ac.jp

2. **Takashi Nodera**, Department of Mathematics, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522, Japan.
mailto:nodera@math.keio.ac.jp