Inexact shift-invert Arnoldi method for evolution equations

Yuka Hashimoto[1] Takashi Nodera[2]

November 16, 2016

Abstract

Linear and nonlinear evolution equations with a first order time derivative, such as the heat equation, the Burgers equation, and the reaction diffusion equation have been used to solve problems in various fields of science. Differential algebraic equations of the first order are derived after space discretization. In the simplest case, the computation of one matrix exponential with a special form is required. In the most complex case, the computation of matrix functions related to its exponentials need to be implemented repeatedly. When computing large matrix functions, the Krylov subspace methods is a viable alternative. The most well-known method is the Arnoldi method, but it may require a number of iterations depending on the condition of the matrix. As a solution to this issue, we propose the Inexact Shift-invert Arnoldi method to do this more efficiently. As the result, the numerical solution

DOI:10.21914/anziamj.v58i0.10766, © Austral. Mathematical Soc. 2016. Published November 16, 2016. 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/10766/0

of evolution equations can be computed efficiently with this method. Pertinent numerical experiments establish the effectiveness of this proposed algorithm.

Subject class: 65F60, 65M22

Keywords: Shift-invert Arnoldi, ϕ-function, exponential integrator

Contents

1 Introduction

1.1 Background

Evolution equations are used in various fields, for example, the heat equation in building physics [22], the Burgers equation in fluid mechanics [16], and the reaction diffusion equation in chemistry [17]. Let $\Omega \subseteq \mathbb{R}^d$ be an open set, $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ be the boundary of Ω, and n_b be the unit normal vector of $\partial\Omega_2$. In addition, we define the time space $[0, T]$, where $T > 0$ is the maximum time we are interested in. We consider the problem defined in $[0, T] \times \overline{\Omega}$, and its solution defined in $\mathcal{V}$, where $\mathcal{V}$ is the norm space contained by $L^2([0, T] \times \overline{\Omega})$. Let $\mathcal{D}$ be a linear or nonlinear differential operator on $\mathcal{V}$, and ξ, η, τ_1 and τ_2 be known functions. The typical example of $\mathcal{D}$ is $\mathcal{D}u = \frac{d^2u}{dx^2} + u\frac{du}{dx}$ for $d = 1$. We explore the initial boundary value problems

$$\begin{cases} \frac{\partial u}{\partial t} = \mathcal{D}u & \text{in } (0, T] \times \Omega, \\ u = \xi & \text{on } \{0\} \times \overline{\Omega}, \\ u = \eta & \text{on } (0, T] \times \partial\Omega_1, \\ \frac{\partial u}{\partial n_b} = \tau_1 u + \tau_2 & \text{on } (0, T] \times \partial\Omega_2. \end{cases} \tag{1}$$

We discretize the equation in terms of space using a finite element method, and derive the differential algebraic equation

$$M\dot{y}(t) = F(t, y(t)), \quad y(0) = v, \tag{2}$$

where $M \in \mathbb{R}^{n\times n}$, and F is a vector valued function. We assume M is invertible. Without a loss of generality, it is assumed that equation (2) is an autonomous system, that is $F = F(y(t))$.

If $\mathcal{D}$ is linear and does not depend on t, F is represented as $F(y) = Ly + c$, where $L \in \mathbb{R}^{n\times n}$ and $c \in \mathbb{R}^n$. We assume L is also invertible as M. Both of them are constants. In this case, the solution of equation (2) is

$$\begin{aligned} y(t) &= e^{tM^{-1}L}M^{-1}v + \int_0^t e^{(t-\tau)M^{-1}L}M^{-1}c\, d\tau \\ &= \phi_0(tM^{-1}L)(v + L^{-1}c) - L^{-1}c\,, \end{aligned} \tag{3}$$

where $\phi_0(z) := e^z$. The solution is obtained through computing the matrix exponential once.

If $\mathcal{D}$ is nonlinear or depends on t, time discretization is also needed for integrating $M^{-1}F(t, y)$ and finding solution $y(t)$. There are various integrators for this kind of problem including classical methods like the explicit and implicit Euler methods [2, pp. 61–65], the Runge–Kutta method [2, pp. 93–104]. The exponential integrator [4, 11, 12, 13] is currently the popular method for solving this problem, because this method is more suitable for stiff problems versus the explicit and implicit Euler methods [13, 14]. In general, at each step, $\phi_k(\Delta t M^{-1}L)$ is required, where

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

and Δt is the step size of time. In this nonlinear case, L is the part which is regarded as “linear” in every time step, such as the Jacobian matrix.

Various methods for computing the matrix exponential and ϕ-functions were introduced [5, 6, 14, 18, 19, 20, 21]. The Krylov subspace methods are efficient, because the matrices usually become large. The most simple and well-known method is the Arnoldi method for the ϕ-function (AP). Hochbruck and Lubich [14, Theorem 5] obtained an error bound for ϕ_0 and ϕ_1. According to this theorem, the error bound does not decrease even if the iteration number becomes larger, if the numerical range of $\Delta t\, M^{-1}L$ is contained in the large disk in the complex plain. This means AP may require a number of iterations if the numerical range of $\Delta t\, M^{-1}L$ is widely distributed. On the other hand, the matrices coming from the spatial discretization of (1) often have a wide numerical range. In order to deal with this difficulty, the Shift-invert Arnoldi method for ϕ-function (SIAP) was proposed by Moret and Novati [19, 21]. According to Novati [21], the SIAP converges independently of the width of the numerical range of $\Delta t\, M^{-1}L$. Moreover, the SIAP is suited to problems like equation (2) which is explored in this article. With this in mind, we propose a new method for computing ϕ-functions based on SIAP, called the Inexact Shift-invert Arnoldi method for ϕ-function (ISIAP). ISIAP is based on SIAP, but it computes ϕ-functions more effectively and guaranteeing the precision of the result. Numerical results also show the effectiveness and the preciseness of our method.

1.2 Notation

The norm is defined as $\|\cdot\| = \|\cdot\|_2$, and the 2-norm condition number of matrix A is defined as $\kappa(A)$. Vector e_j represents the jth column of identity matrix I. Moreover, let $\mathbb{C}^- := \{z \in \mathbb{C} \mid \Re(z) < 0\}$, and $W(A) := \{u^*Au \mid u \in \mathbb{C}^n,\ \|u\| = 1\}$ be the numerical range of $n \times n$ matrix A.

2 Numerical methods for evolution equations

2.1 Exponential integrator

At the ith step, the exponential integrator rearranges F as

$$F(y) = L_i y(t) + n(y), \tag{4}$$

where L_i is the pseudo linear part of the ith step. For example, $L_i = \frac{\partial}{\partial y}F(y(t_0))$, $L_i = \frac{\partial}{\partial y}F(y(t_{i-1}))$, and $n(y) = F(y) - L_i y$ for $t \in (t_i, t_{i+1}]$. The solution is approximated at the $(i+1)$th step as

$$y(t_i+\tau) = e^{\tau M^{-1}L_{i+1}} y_i + \int_0^\tau e^{(\tau-\sigma)M^{-1}L_{i+1}} M^{-1} n(y(t_i+\sigma))\,d\sigma\,, \quad \tau \in (0, \Delta t], \tag{5}$$

where $t_i = i\Delta t$ $(i = 0, \ldots, N)$, $t_N = T$, and y_i is the approximation of $y(t_i)$ at the ith step. If the nodes $0 \leqslant c_1 < c_1 < \cdots < c_s \leqslant 1$ are chosen for the approximation of the integral in (5) by a quadrature formula, then the following scheme of the one-step method is obtained [13]. For $1 \leqslant k \leqslant s$,

$$\begin{aligned} Y_{ik} &= \phi_0(c_k \Delta t M^{-1} L_{i+1}) y_i + \Delta t \sum_{l=1}^{k-1} a_{kl}(\Delta t M^{-1} L_{i+1}) M^{-1} n_i(Y_{il}), \\ y_{i+1} &= \phi_0(\Delta t M^{-1} L_{i+1}) y_i + \Delta t \sum_{k=1}^{s} b_k(\Delta t M^{-1} L_{i+1}) M^{-1} n_i(Y_{ik}), \end{aligned} \tag{6}$$

where a_{kl} and b_k are coefficients which consist of ϕ-functions. Let $p \in \mathbb{N}$. If the coefficients are chosen with satisfying the condition shown by Hochbruck and Ostermann [13, Theorem 2.22, Table 2.3], then the approximation obtained from equations (6) converges with order p.

The approximation of the simplest case of $s = 1$, is

$$\begin{aligned} y_{i+1} &= \phi_0(\Delta t M^{-1} L_{i+1}) y_i + \Delta t \phi_1(\Delta t M^{-1} L_{i+1}) M^{-1} n(y_i) \\ &= y_i + \Delta t \phi_1(\Delta t M^{-1} L_{i+1}) M^{-1} F(y_i). \end{aligned} \tag{7}$$

In the case of $s = 2$, the coefficients are, for example, $c_1 = 0$, $c_2 = 1$, $a_{21} = \phi_1$, $b_1 = \phi_1 - 2\phi_3$, and $b_2 = 2\phi_3(z)$. Various ways of choosing a_{kl}, b_k, and c_k have been suggested. Hochbruck et al. [13] discuss more details.

The multi-step method is also mentioned by Hochbruck and Ostermann [13]. The approximation scheme of the r-step method is

$$y_{i+1} = \phi_0(\Delta t M^{-1} L_{i+1}) y_i + \Delta t \sum_{k=1}^{r-1} \gamma_k(\Delta t M^{-1} L_{i+1}) M^{-1} \nabla^k N_i \,, \tag{8}$$

where $N_i := n(y_i)$, and $\nabla^k N_i$ and $\gamma_k(z)$ are defined recursively by

$$\nabla^0 N_i := N_i \,, \quad \nabla^{k+1} N_i := \nabla^k N_i - \nabla^k N_{i-1} \,,$$

$$\gamma_0(z) = \phi_1(z), \quad z\gamma_k(z) + 1 = \sum_{l=0}^{k-1} \frac{1}{k-l} \gamma_l(z).$$

The approximation of the simplest case of $r = 1$ is equation (7). In the case of $r = 2$, one of the approximations is

$$y_{i+1} = y_i + \Delta t \phi_1(\Delta t M^{-1} L_{i+1}) F(y_i) - \Delta t \frac{2}{3} \phi_2(\Delta t M^{-1} L_{i+1}) [n(u_i) - n(u_{i-1})].$$

2.2 Shift-invert Arnoldi method (SIAP)

In this subsection, the SIAP is used to compute $\phi_k(tM^{-1}L)M^{-1}v$, in the same manner as the ϕ-functions that appear in equations (3), (6) and (8). In the case of equation (3), v is replaced by $M(v + L^{-1}c)$.

Let $\beta = \|M^{-1}v\|$. Then compute the m-step Arnoldi process for the shift and invert matrix of $M^{-1}L$, $(I - \gamma M^{-1}L)^{-1} = (M - \gamma L)^{-1}M$, where $\gamma > 0$ is a shift. From this computation with the initial vector $v_1 = M^{-1}v/\beta$, the relations

$$\begin{aligned}
h_{j+1,j} v_{j+1} &= (M - \gamma L)^{-1} M v_j - \sum_{k=1}^{j} h_{k,j} v_k \,, \\
h_{k,j} &= v_k^* \left[(M - \gamma L)^{-1} M v_j - \sum_{l=1}^{k-1} h_{l,j} v_l \right], \\
h_{j+1,j} &= \left\| (M - \gamma L)^{-1} M v_j - \sum_{k=1}^{j} h_{k,j} v_k \right\| \qquad (j = 1, \ldots, m),
\end{aligned}$$

are derived. This relation is expressed with matrices as

$$(M - \gamma L)^{-1} M V_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \,, \tag{9}$$

$$V_m^T (M - \gamma L)^{-1} M V_m = H_m \,, \tag{10}$$

where $V_m = [v_1 \ \cdots \ v_m]$ is an $n \times m$ matrix whose columns are orthonormal, and H_m is an $m \times m$ upper Hessenberg matrix. $\phi_k(tM^{-1}L)M^{-1}v$ can be regarded as $\psi_k\left([(I - \gamma M^{-1}L)^{-1}]^{-1}\right) M^{-1}v$, the function of $(I - \gamma M^{-1}L)^{-1}$, where $\psi_k(z) := \phi_k\left(t(1-z)/\gamma\right)$. Therefore, if H_m is invertible, then the matrix function

$$\begin{aligned}\phi_k(tM^{-1}L)M^{-1}v &\approx \beta V_m V_m^T \psi_k\left(\left[(I-\gamma M^{-1}L)^{-1}\right]^{-1}\right) M^{-1}v\\ &\approx \beta V_m \psi_k\left(\left[V_m^T(M-\gamma L)^{-1}MV_m\right]^{-1}\right) e_1\\ &= \beta V_m \psi_k(H_m^{-1})e_1\\ &=: V_m u_m^{\text{SI}}(t).\end{aligned}$$

The following proposition regarding the error bound of this approximation was proven by Novati [21, Proposition 12].

Proposition 1. *Let* $0 \leqslant \theta < 0.48124$, *and* $S_\theta := \{z \in \mathbb{C}^- \mid |\arg(-z)| \leqslant \theta\}$. *If* $W(M^{-1}L) \subseteq S_\theta$ *and* $t/\gamma = (m+k)/\cos\theta$, *then the following error bound holds:*

$$\|\phi_k(tM^{-1}L)v - V_m u_m^{SI}(t)\| \leqslant 11\, C\rho(\theta)^m, \tag{11}$$

where

$$\rho(\theta) := \left(1+\sqrt{2(1-\cos\theta)}\right)\frac{\cos\theta}{4\cos\theta-2}\,\frac{\pi}{\pi-\theta},$$

and $1 \leqslant C \leqslant 11.08$.

Note that ρ in the right-hand side of inequality (11) only depends on angle θ. Thus, Proposition 1 implies that if γ is chosen, satisfying $t/\gamma = (m+k)/\cos\theta$, then the convergence does not depend on the width of the numerical range of $tM^{-1}L$.

3 Inexact Shift-invert Arnoldi method (ISIAP)

In this section the ISIAP is used to compute $\phi_k(tM^{-1}L)M^{-1}v$. Throughout this section we assume $W\left((I-\gamma M^{-1}L)^{-1}\right) \subseteq \mathbb{C}^-$. Computing this with AP requires a product of $M^{-1}L$ and vector v_m at the mth step. It is necessary to solve a linear equation $Mx_m = Lv_m$ for x_m. The computation with the SIAP also requires solving another linear equation $(M - \gamma L)x_m = Mv_m$. The computational costs for one step are approximately the same for both. Because SIAP converges independently of the width of the numerical range of $tM^{-1}L$, it is the efficient choice for computing ϕ-functions. However, even with the SIAP, linear equations must be solved at every step and this results in a high computational cost. An attempt is made to reduce this, by solving the linear equation inexactly with an iterative method. Any iterative methods, for example, BiCGStab or GMRES, are viable options.

For $j = 1, \dots, m$, let $\tilde{x}_j$ be the inexact solution of the linear equation $(M - \gamma L)x_j = Mv_j$, and $F_m := [f_1 \ \cdots \ f_m]$, where $f_j := x_j - \tilde{x}_j$ is the error vector for solving the linear equation, and let $R_m := [r_1^{\text{sys}} \ \cdots \ r_m^{\text{sys}}]$, where $r_j^{\text{sys}} := Mv_j - (M - \gamma L)\tilde{x}_j$ is the residual vector for solving the linear equation. The following relation is derived by computing the m-step Arnoldi process for $(M - \gamma L)^{-1}M$ in the same way as Section 2.2. However, in this case, the linear equations must be solved inexactly at every step.

$$(M - \gamma L)^{-1}MV_m - F_m = V_mH_m + h_{m+1,m}v_{m+1}e_m^T\,, \tag{12}$$

$$\begin{aligned} MV_m - R_m &= (M - \gamma L)V_mH_m \\ &\qquad + h_{m+1,m}(M - \gamma L)v_{m+1}e_m^T\,, \end{aligned} \tag{13}$$

where V_m is the $n \times m$ matrix whose columns are orthonormal, and H_m is an $m \times m$ upper Hessenberg matrix. The matrices V_m and H_m in equation (12) and (13) are different matrices from equation (9) and (10). If H_m is invertible, then

$$\phi_k(tM^{-1}L)M^{-1}v \approx \beta V_m\psi_k(H_m^{-1})e_1 =: V_mu_m^{\text{ISI}}(t). \tag{14}$$

The error of this approximation, using Cauchy's integral formula, is

$$\begin{aligned} E_m &= \psi_k\left(M^{-1}(M - \gamma L)\right)M^{-1}v - \beta V_m\psi_k(H_m^{-1})e_1 \\ &= \frac{1}{2\pi i}\int_\Gamma \psi_k(\lambda)\left\{\left[\lambda I - M^{-1}(M - \gamma L)\right]^{-1}M^{-1}v \right. \\ &\qquad\qquad \left. -\beta V_m(\lambda I - H_m^{-1})^{-1}e_1\right\}\,d\lambda \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2\pi i}\int_\Gamma \psi_k(\lambda)\left\{[\lambda M - (M-\gamma L)]^{-1} v - \beta V_m(\lambda I - H_m^{-1})^{-1} e_1\right\}\, d\lambda \\
&= \frac{1}{2\pi i}\int_\Gamma \psi_k(\lambda) e_m^{\text{lin}}\, d\lambda\,, \qquad (15)
\end{aligned}
$$

where Γ is a contour enclosing the eigenvalues of $M^{-1}(M-\gamma L)$ and H_m^{-1}. The vector $\beta V_m\left(\lambda I - H_m^{-1}\right)^{-1} e_1$ is the approximation of the solution of $[\lambda M - (M-\gamma L)]\, x = v$, and e_m^{lin} represents the error of this approximation for the linear equation. The residual r_m^{lin} of this approximation for the linear equation is

$$
\begin{aligned}
r_m^{\text{lin}} &= v - [\lambda M - (M-\gamma L)]\, \beta V_m \left(\lambda I - H_m^{-1}\right)^{-1} e_1 \\
&= v - \beta\lambda M V_m \left(\lambda I - H_m^{-1}\right)^{-1} e_1 + \beta \left[M V_m H_m^{-1} \right. \\
&\qquad \left. + R_m H_m^{-1} - h_{m+1,m}(M-\gamma L) v_{m+1} e_m^T H_m^{-1}\right] \left(\lambda I - H_m^{-1}\right)^{-1} e_1 \\
&= v - \beta M V_m \left(\lambda I - H_m^{-1}\right)\left(\lambda I - H_m^{-1}\right)^{-1} e_1 \\
&\qquad + \left[\beta R_m H_m^{-1} - \beta h_{m+1,m}(M-\gamma L) v_{m+1} e_m^T H_m^{-1}\right] \left(\lambda I - H_m^{-1}\right)^{-1} e_1 \\
&= \left[\beta R_m H_m^{-1} - \beta h_{m+1,m}(M-\gamma L) v_{m+1} e_m^T H_m^{-1}\right] \left(\lambda I - H_m^{-1}\right)^{-1} e_1\,.
\end{aligned}
$$

Replacing e_m^{lin} by r_m^{lin} in equation (15), the generalized residual $r_{\phi,m}^{\text{real}}$ [15] of approximating $\phi_k(tM^{-1}L)M^{-1}v$ is

$$
\begin{aligned}
r_{\phi,m}^{\text{real}} &= -\beta h_{m+1,m}(M-\gamma L) v_{m+1} e_m^T H_m^{-1} \psi_k(H_m^{-1}) e_1 \\
&\qquad + \beta R_m H_m^{-1} \psi_k(H_m^{-1}) e_1\,. \qquad (16)
\end{aligned}
$$

In order to evaluate equation (16), the following proposition is used.

Proposition 2. *Let* $f(z) := \beta z^{-1}\psi(z^{-1})$. *If*

$$
W(H_m) \subseteq \mathbb{C}^-, \qquad (17)
$$

then there exist $K > 0$ *and* $0 < \lambda < 1$ *which do not depend on* m *and satisfy*

$$
\left|[f(H_m)]_{i,j}\right| \leqslant K\lambda^{i-j} \quad (i \geqslant j). \qquad (18)
$$

Proof: Because of the boundedness of $W(H_m)$ and the assumption, there is a simply connected compact Jordan region $\mathcal{F}$ which satisfies condition $W(H_m) \subseteq \mathcal{F} \subseteq \mathbb{C}^-$. Let $\bar{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$. Due to Riemann's mapping theorem, there is a biholomorphism $\Phi : \bar{\mathbb{C}} \backslash \mathcal{F} \mapsto \{w \in \bar{\mathbb{C}} \mid |w| > \rho\}$ which satisfies condition $\Phi(\infty) = \infty$, $\lim_{z\to\infty}(\Phi(z)/z) = 1$. $\rho > 0$ is denoted as a logarithmic capacity of $\mathcal{F}$. Due to Carathéodory's Theorem [3], Φ is extended to $\overline{\bar{\mathbb{C}} \backslash \mathcal{F}}$ as a homeomorphism. Let Ψ be the inverse of Φ. Because of the continuity of Ψ, there exists $R_0 > \rho$ in such a way that the Jordan region of $\Psi\left(\{w \in \bar{\mathbb{C}} \mid |w| = R_0\}\right)$ does not include $\{0\}$. Let $I(C_{R_0})$ be this Jordan region. Since f is regular in $I(C_{R_0})$, and H_m is an upper Hessenberg matrix, the proposition follows that of Benzi's Theorem [1, Theorem 11]. ♠

This proposition means that if condition (17) is satisfied, then the entries of $f(H_m)$ decays exponentially along the diagonal. If $\|r_m^{\mathrm{sys}}\| \leqslant \delta$, $(\delta > 0)$, then the upper bound of the first term of equation (16) is estimated as

$$\begin{aligned}
&\left|h_{m+1,m}\left[e_m^T f(H_m) e_1\right]\right| \, \|(M-\gamma L)v_{m+1}\| \\
&\quad\leqslant |h_{m+1,m}| \left|[f(H_m)]_{m,1}\right| \, \|M-\gamma L\| \, \|v_{m+1}\| \\
&\quad\leqslant |h_{m+1,m}| \, \|(M-\gamma L)\| K\lambda^{m-1} \\
&\quad\leqslant \|(M-\gamma L)^{-1} M v_m - f_m - h_{1,n}v_1 - \cdots - h_{m,m}v_m\| \, \|M-\gamma L\| K\lambda^{m-1} \\
&\quad\leqslant (\|(M-\gamma L)^{-1} M v_m\| + \|f_m\|)\|M-\gamma L\| K\lambda^{m-1} \\
&\quad\leqslant (\|M\| + \|r_m^{\mathrm{sys}}\|)\|(M-\gamma L)^{-1}\| \, \|M-\gamma L\| K\lambda^{m-1} \\
&\quad\leqslant (\|M\| + \delta)\kappa(M-\gamma L) K\lambda^{m-1}.
\end{aligned}$$

Because $0 < \lambda < 1$, the first term of equation (16) becomes smaller as m becomes larger.

Remark 3. If $F_m = O$, then $W(H_m)$ satisfies the condition $W(H_m) \subseteq W((I-\gamma M^{-1}L)^{-1}) \subseteq \mathbb{C}^-$. Thus, if H_m does not satisfy condition (17), then a smaller δ should be chosen to minimize the error in solving the linear equation, or a smaller γ should be chosen to separate $W\left((I-\gamma M^{-1}L)^{-1}\right)$ from the origin in the complex plain.

For the second term of equation (16), the following theorem is deduced.

Theorem 4. *Let* $[f(H_m)]_{i,j} =: g^m_{i,j}$, *and let* $\mathrm{tol}_\phi > 0$ *be the convergence threshold for computing the* ϕ*-function. If*

$$\|r^{\mathrm{sys}}_1\| \leqslant \frac{\mathrm{tol}_\phi}{m^{\max}\|f(H_m)e_1\|}\,, \tag{19}$$

$$\|r^{\mathrm{sys}}_j\| \leqslant \frac{|g^m_{1,1}|}{|g^m_{j-1,1}|}\|r^{\mathrm{sys}}_1\| \qquad (2 \leqslant j \leqslant m), \tag{20}$$

then

$$\|R_m f(H_m)e_1\| \leqslant \mathrm{tol}_\phi\,.$$

Proof: Let $m^{\max}$ be the largest number of iterations. Based on the above assumptions (19), (20) and Proposition 2, we derive the upper bound

$$\begin{aligned}
\|R_m f(H_m)e_1\| &\leqslant |g^m_{1,1}|\,\|r^{\mathrm{sys}}_1\| + |g^m_{2,1}|\,\|r^{\mathrm{sys}}_2\| + \cdots + |g^m_{m,1}|\,\|r^{\mathrm{sys}}_m\| \\
&\leqslant |g^m_{1,1}|\,\|r^{\mathrm{sys}}_1\| + |g^m_{2,1}|\frac{|g^m_{1,1}|}{|g^m_{1,1}|}\|r^{\mathrm{sys}}_1\| + |g^m_{3,1}|\frac{|g^m_{1,1}|}{|g^m_{2,1}|}\|r^{\mathrm{sys}}_1\| \\
&\qquad + \cdots + |g^m_{m,1}|\frac{|g^m_{1,1}|}{|g^m_{m-1,1}|}\|r^{\mathrm{sys}}_1\| \quad \text{(because of (20))} \\
&= |g^m_{1,1}|\,\|r^{\mathrm{sys}}_1\| \left(1 + \frac{|g^m_{2,1}|}{|g^m_{1,1}|} + \frac{|g^m_{3,1}|}{|g^m_{2,1}|} + \cdots + \frac{|g^m_{m,1}|}{|g^m_{m-1,1}|}\right) \\
&\leqslant \|f(H_m)e_1\|\,\|r^{\mathrm{sys}}_1\|\,(1 + \lambda + \cdots + \lambda) \quad \text{(because of (18))} \\
&\leqslant \|f(H_m)e_1\|\,\|r^{\mathrm{sys}}_1\| \cdot m^{\max} \\
&\leqslant \mathrm{tol}_\phi \quad \text{(because of (19))}.
\end{aligned}$$

♠

The right-hand side of inequality (20) becomes larger as m becomes larger because of Proposition 2. Thus, Theorem 4 implies that the larger m becomes,

Algorithm 1 Inexact Shift-invert Arnoldi method (ISIAP)

```
Require: $L, M \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^n$, $t \in (0,T]$, $\gamma > 0$, $\delta > 0$, $\mathrm{tol}_\phi > 0$, $m^{\max}$
Ensure: $\beta V_m \psi_k(H_m^{-1})e_1$ such that $\|r_{\phi,m}^{\mathrm{real}}\| \leqslant \mathrm{tol}_\phi$
$\beta = \|M^{-1}v\|$, $v_1 = M^{-1}v/\beta$
$\mathrm{tol}_1^{\mathrm{sys}} = \mathrm{tol}_\phi / (m^{\max}\|f_{m(i)}^i\|)$
for $m = 1, 2, \dots, m^{\max}$ do
    Compute $\tilde{x}$ such that $\|Mv_m - (M - \gamma L)\tilde{x}\| \leqslant \mathrm{tol}_m^{\mathrm{sys}}$
    for $l = 1, 2, \dots, m$ do
        $h_{l,m} = \tilde{x}^T v_l$
        $\tilde{x} = \tilde{x} - h_{l,m} v_k$
    end for
    $h_{m+1,m} = \|\tilde{x}\|$, $v_{m+1} = \tilde{x}/h_{m+1,m}$
    $f_m^{i+1} = H_m^{-1}\psi_k(H_m^{-1})e_1$
    $r = |h_{m+1,m}(f_m^{i+1})_1| \, \|(M - \gamma L)v_{m+1}\|$
    $\mathrm{tol}_{m+1}^{\mathrm{sys}} = \min\{\mathrm{tol}_1^{\mathrm{sys}} \, |(f_m^{i+1})_1| / |(f_m^{i+1})_m|, \delta\}$
    if $r \leqslant \mathrm{tol}_\phi$ then
        $m(i+1) = m$
        $y_{m(i+1)}(t) = V_{m(i+1)}\psi_k(H_{m(i+1)}^{-1})e_1$, break
    end if
end for
```

the solution of linear equation $(M - \gamma L)x_m = Mv_m$ becomes more inexact, and the computational cost decreases compared to the SIAP. However, if the linear equations are solved, satisfying inequalities (19) and (20), then the second term of equation (16) is no longer an issue. In this scenario, the first term of equation (16), $r_{\phi,m}^{\mathrm{comp}}$, is used as the stopping criterion for the convergence of ISIAP.

Remark 5. For ϕ_0, the standard residual dealt by Gang et al. [5],

$$\begin{aligned} r_{\phi,m}^{\mathrm{real}} &= -\frac{\beta}{\gamma} h_{m+1,m} \left[e_m^T H_m^{-1} \psi_0(H_m^{-1}) e_1 \right] (I + \gamma A) v_{m+1} \\ &\quad + \frac{\beta}{\gamma} R_m H_m^{-1} \psi_0(H_m^{-1}) e_1, \end{aligned} \tag{21}$$

is available. The residual (21) is the same as the residual (16) up to the constant factor $1/\gamma$. Therefore, we replace residual (16) by residual (21) for computing ϕ_0.

Remark 6. In practical computation, the values depending on m in equations (19) and (20) are unavailable in advance. Thus, for computing equation (3), we use the approximation

$$\begin{aligned}\|f(H_m)e_1\|\,\|r_1^{\text{sys}}\| &\approx \|\beta V_m^T M^{-1}(M-\gamma L)V_m\psi_k(H_m^{-1})e_1\|\,\|r_1^{\text{sys}}\| \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{(because of (12))} \\ &\approx \|M^{-1}(M-\gamma L)y(t)\|\,\|r_1^{\text{sys}}\| \quad \text{(because of (14))} \\ &\approx \|M^{-1}(M-\gamma L)(v+L^{-1}c)\|\,\|r_1^{\text{sys}}\|.\end{aligned}$$

The matrices and vectors in the m dimensional Krylov subspace are approximated with ones in the original space. We also approximate $y(t)$ with $y(0)$. For computing equation (6) and (8) for the exponential integrator at the $(i+1)$th step, $f(H_m)e_1$ is replaced with the ones in the largest Krylov subspace at the ith step. Concerning inequality (20), K and λ in inequality (18) do not depend on m, so we use the approximation

$$|g_{1,1}^m| \approx |g_{1,1}^{j-1}|, \quad |g_{1,j-1}^m| \approx |g_{1,j-1}^{j-1}| \qquad (2 \leqslant j \leqslant m).$$

In summary, we propose Algorithm 1 for ϕ-functions in equation (6) and (8), where $(f_m)_j$ is the jth element of f_m. Güttel [8] and Göckler [7] discussed the method for choosing an appropriate γ, and Hashimoto and Nodera [9] showed the way of confirming whether the shift is suitable for ISIAP or not. The linear equation in the fourth line of the algorithm is solved by an iterative method, and the convergence of its solution is judged by its residual. Therefore, it is easy to ensure that the residual of the solution of the linear equation satisfies the required condition. H_m^{-1} in the tenth line is a small matrix, so it is computed by a direct method inexpensively. After computing H_m^{-1}, $\psi_k(H_m^{-1})e_1$ is also be computed by a direct method, like the scaling and squaring methods [10]. The algorithm for computing equation (3) can be obtained through replacing the second line with $\text{tol}_1^{\text{sys}} = \text{tol}_\phi \,/\, [m^{\max}\|M^{-1}(M-\gamma L)(v+L^{-1}c)\|]$.

4 Numerical experiments

In this section, a few typical numerical experiments are implemented. These experiments are in a collection of problems to illustrate the effectiveness of the ISIAP. All numerical computations of these tests were done with MATLAB 2015a on an Intel(R) Xeon(R) E3-1270 V2 processor with a CPU of 3.5 GHz with a Ubuntu14.04LTS operating system.

The Galerkin method with unstructured first order triangle elements and linear weight functions, were used to discretize the problems. After the discretization, the BiCGStab algorithm [23] with an ILU(0) preconditioner were applied to solve $(M-\gamma L)x_m = Mv_m$, or $Mx_m = Lv_m$ in every iteration in the AP, SIAP, and ISIAP. For the AP and SIAP, the linear equation was solved with a residual tolerance of 10^{-14}.

Example 7. The convection diffusion equation in region $\Omega = ((-1.5, 1.5) \times (-1, 1)) \subseteq \mathbb{R}^2$ is first described as

$$\begin{cases} \rho c_\nu \frac{\partial u}{\partial t} = \lambda \Delta u - 5\frac{\partial u}{\partial x_1} & \text{in } (0, T] \times \Omega, \\ u = 300 & \text{on } \{0\} \times \Omega, \\ -\lambda \frac{\partial u}{\partial n} = \alpha(u - 280) & \text{on } (0, T] \times \partial\Omega_1, \\ -\lambda \frac{\partial u}{\partial n_b} = -1 & \text{on } (0, T] \times \partial\Omega_2, \end{cases}$$

where $\partial\Omega_2 = \{0.5\} \times [-1, 1]$, $\partial\Omega_1 = \partial\Omega \setminus \partial\Omega_1$, $c = [5\ 0]$, $\rho = 1.29$, $c_\nu = 1000$, $\lambda = 0.025$ and $\alpha = 9.3$. After the discretization, equation (2) with $F(y) = Ly + c$ is obtained. The solution is obtained through computing equation (3). Equation (3) is computed with the AP, SIAP, and ISIAP. We compare the CPU time, iteration numbers and relative errors. Table 1 details the results. The relative residual tolerance tol_ϕ for computing $\phi_0(tM^{-1}L)(v + L^{-1}c)$ is 10^{-8}, and $t = 300$. $r^{\text{comp}}_{\phi,m}$ is replaced with $r^{\text{comp}\,\prime}_{\phi,m} := r^{\text{comp}}_{\phi,m} / \|M^{-1}(v + L^{-1}c)\|$ to obtain relative residuals. For the SIAP and the ISIAP, $\gamma = 5$ is used, and for the ISIAP, $\delta = 10^{-2}$ and $m^{\max} = 100$ are used. The solutions with AP with a residual tolerance 10^{-14} are used as the exact solution to estimate the relative errors. The results suggests that the larger n becomes, the more iterations

Table 1: Example 7, Comparison of ISIAP, SIAP, and AP.

n	Algorithm	CPU (sec)	Iterations	Relative error
1925	AP	0.30	106	2.1e−09
	SIAP	0.18	50	6.1e−09
	ISIAP	0.12	50	6.1e−09
7561	AP	1.72	183	8.2e−09
	SIAP	0.53	54	1.9e−07
	ISIAP	0.34	54	1.9e−07
29969	AP	15.44	339	3.3e−08
	SIAP	2.18	55	1.6e−07
	ISIAP	1.29	55	1.6e−07

AP needs. This is because $W(tM^{-1}L)$ becomes larger as n becomes larger. On the other hand, the number of iterations the SIAP and the ISIAP needed are almost the same in all n. Moreover, the ISIAP is the fastest of all three algorithms, whereas there is no noticeable difference in terms of relative error. Figure 1 shows the relationship between the number of iterations and the relative residuals. The real relative residual ${r^{\text{real}}_{\phi,m}}' := r^{\text{real}}_{\phi,m}/\|M^{-1}(v + L^{-1}c)\|$ decreased until it reached tol_ϕ, but it stopped decreasing after this point. This means that the linear equation can be solved efficiently at each Arnoldi step. On the other hand, the computing residual $r^{\text{comp}\,\prime}_{\phi,m}$ decreased even after it reached tol_ϕ. Moreover, the behavior of ${r^{\text{real}}_{\phi,m}}'$ and $r^{\text{comp}\,\prime}_{\phi,m}$ are the same before they reach tol_ϕ. Thus, $r^{\text{comp}\,\prime}_{\phi,m}$ is appropriate for the stopping criterion. Table 2 shows the residual tolerance for solving linear equations at each Arnoldi step for $n = 29969$. We see that the exactness needed to obtain a solution for the linear equation decreases as m becomes larger. Figure 2 shows the solution computed with the ISIAP, $n = 29969$. The exactness of the computing is illustrated here.

Example 8. The second test problem is Burgers equation in region $\Omega =$

Table 2: Example 7, $n = 29969$: The residual tolerance $\mathrm{tol}_m^{\mathrm{sys}}$ for solving linear equations at each Arnoldi step m.

m	$\mathrm{tol}_m^{\mathrm{sys}}$	m	$\mathrm{tol}_m^{\mathrm{sys}}$	m	$\mathrm{tol}_m^{\mathrm{sys}}$
1	7.4e−11	26	7.8e−10	51	1.5e−04
2	5.2e−10	27	1.1e−09	52	6.4e−05
3	4.9e−10	28	2.1e−09	53	5.2e−04
4	5.3e−10	29	6.2e−09	54	1.4e−04
5	5.4e−10	30	3.8e−08	55	1.5e−03

Figure 1: Example 7, $n = 29969$, Iterations versus $\|r_{\phi,m}^{\mathrm{real}}{}'\|$ and $\|r_{\phi,m}^{\mathrm{comp}}{}'\|$.

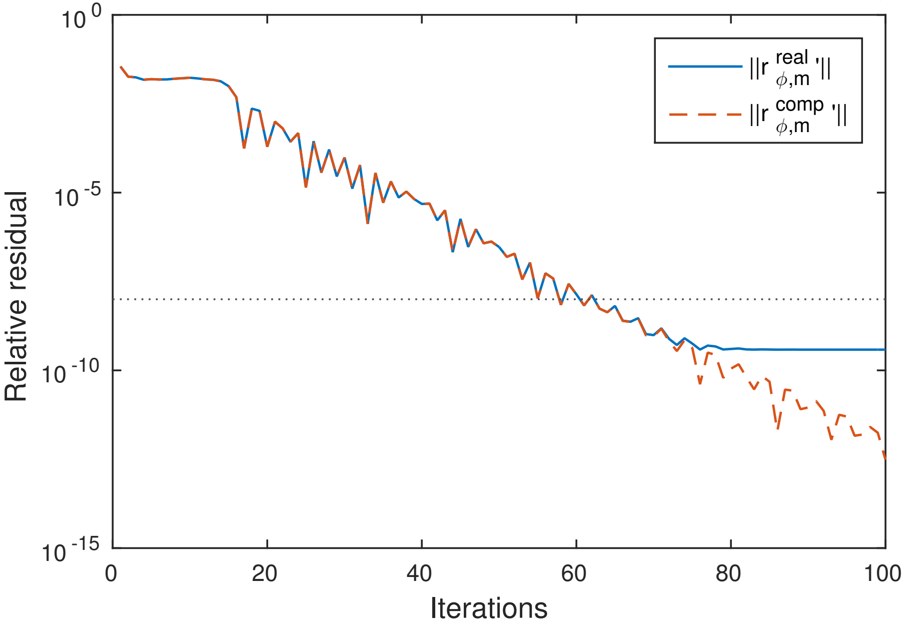
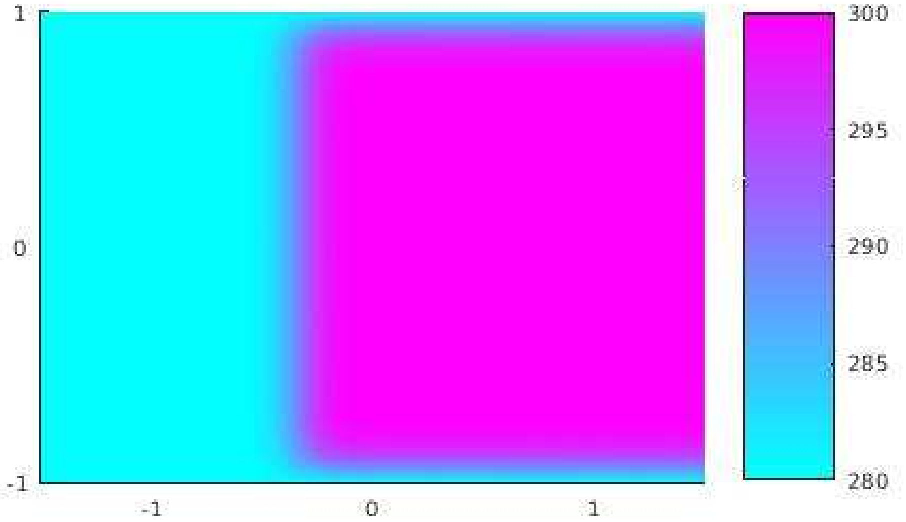


Figure 2: Example 7, $n = 29969$: Computational solution.



$(0,1) \times (0,1) \subseteq \mathbb{R}^2$

$$\begin{cases} \frac{\partial u}{\partial t} = u\frac{\partial u}{\partial x_1} + v\frac{\partial u}{\partial x_2} + \frac{1}{\mathrm{Re}}\Delta u & \text{in } (0,T] \times \Omega, \\ \frac{\partial v}{\partial t} = u\frac{\partial v}{\partial x_1} + v\frac{\partial v}{\partial x_2} + \frac{1}{\mathrm{Re}}\Delta v & \text{in } (0,T] \times \Omega, \\ u = u^{\mathrm{anal}}(0,x), \quad v = v^{\mathrm{anal}}(0,x) & \text{on } \{0\} \times \Omega, \\ u = u^{\mathrm{anal}}(t,x), \quad v = v^{\mathrm{anal}}(t,x) & \text{on } (0,T] \times \partial\Omega, \end{cases}$$

where $\mathrm{Re} = 100$, $u^{\mathrm{anal}} = 3/4 - 1/[4 + 4e^{\mathrm{Re}(-t-4x_1+4x_2)/32}]$ and $v^{\mathrm{anal}} = 3/4 + 1/[4 + 4e^{\mathrm{Re}(-t-4x_1+4x_2)/32}]$. The analytic solution of this problem is u^{anal} and v^{anal} [16]. After the discretization, equation (2) is obtained with $F(y) = Ly + Q(y)y + n(t)$. To show the effectiveness of the exponential integrator, the solution computed with the Semi Implicit Euler (SIE) and Exponential Integrator of $s = 1$ and $r = 1$ (EI), are compared. For SIE, we use the scheme

$$B\frac{y_{i+1} - y_i}{\Delta t} = L_{i+1}y_{i+1} + n(t_i),$$

$$(B - \Delta t L_{i+1})(y_{i+1} - y_i) = F(y_i), \tag{22}$$

where $L_i = L + Q(y_{i-1})$. The linear equation (22) is solved with the BiCGStab with an ILU(0) preconditioner. The same pseudo linear part L_i is used for EI. The cost of computing the exact Jacobian matrix is prohibitively high. Therefore, the approximation of the Jacobian-vector product is often used for EI [4]. Unfortunately, using this approximation requires the evaluation of F for many points when the ISIAP is used. For problems like equation (2), this evaluation is also costly. This suggests that constructing the explicit pseudo linear part during each step is important for the effective use of ISIAP. For this reason, the pseudo linear part $L_i = L + Q(y_{i-1})$ is set to have one function evaluation for each step. Moreover, an ISIAP of $\gamma = 10^{-2}$, $m^{\max} = 100$, $\delta = 10^{-2}$ are used to compute ϕ-functions in EI. A residual tolerance of 10^{-8} is chosen for the ϕ-functions of the EI and the linear equation of SIE at each time step. Figure 3 and Figure 4 show the relative errors at matrix dimension $n = 1234,\ 5090,\ 20674,\ 83330$ and time step $\Delta t = 10^{-1},\ 10^{-2}$ for computing the solution of $t = 1$. The accuracy of SIE worsens as n becomes larger. On the other hand, that of EI improves as n becomes larger. Next, the ISIAP, SIAP, and AP are compared, for computing ϕ-functions in the EI. The same γ, $m^{\max}$, δ, and residual tolerance are used for ϕ-functions. The time step is set to $\Delta t = 10^{-2}$. Table 3 shows the CPU time and the relative error of each algorithm for computing the solution at $t = 1$. ISIAP is the fastest for all n, while the relative error is more or less the same for all algorithms.

Example 9. The next test problem explores using the reaction-diffusion Brusselator equation in region $\Omega = (-1, 1) \times (-1, 1) \subseteq \mathbb{R}^2$

$$\begin{cases} \frac{\partial u}{\partial t} = B + u^2 v - (A+1)u + \alpha \Delta u & \text{in } (0, T] \times \Omega, \\ \frac{\partial v}{\partial t} = Au - u^2 v + \alpha \Delta v & \text{in } (0, T] \times \Omega, \\ u = u_0\,, \quad v = 1 & \text{on } \{0\} \times \Omega, \\ u = v = 0 & \text{on } (0, T] \times \partial\Omega_1, \\ \frac{\partial u}{\partial n_b} = \frac{\partial v}{\partial n_b} = 0 & \text{on } (0, T] \times \partial\Omega_2, \end{cases}$$

where $A = B = 1$, $\partial\Omega_1 = [-1, 1] \times \{-1\}$, $\partial\Omega_2 = \partial\Omega \setminus \partial\Omega_1$, and u_0 is the $\{0, 2\}$-value function shown at $t = 0$ in Figure 5. The discretization results

Figure 3: Example 8, The relative error of SIE and EI of $\Delta t = 10^{-1}$.

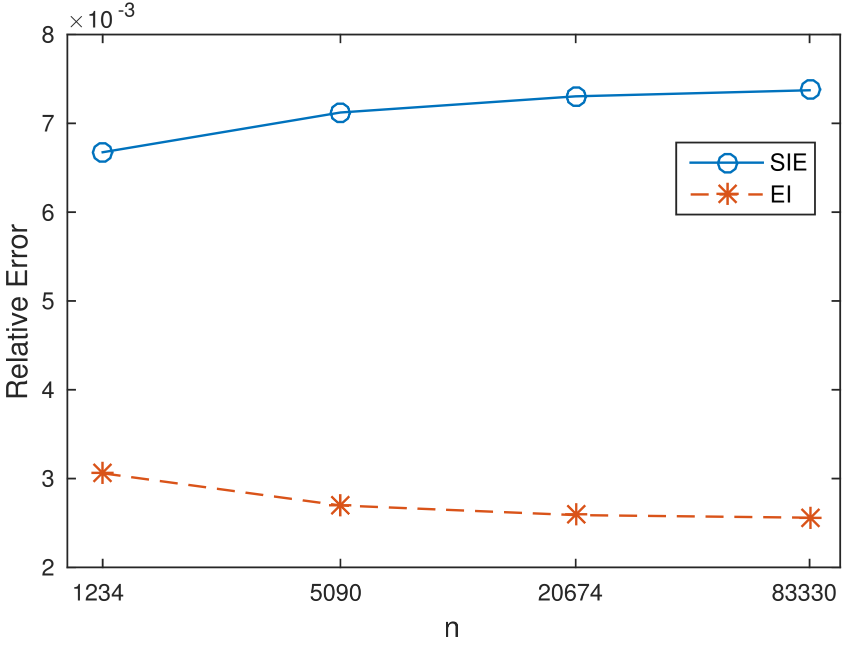
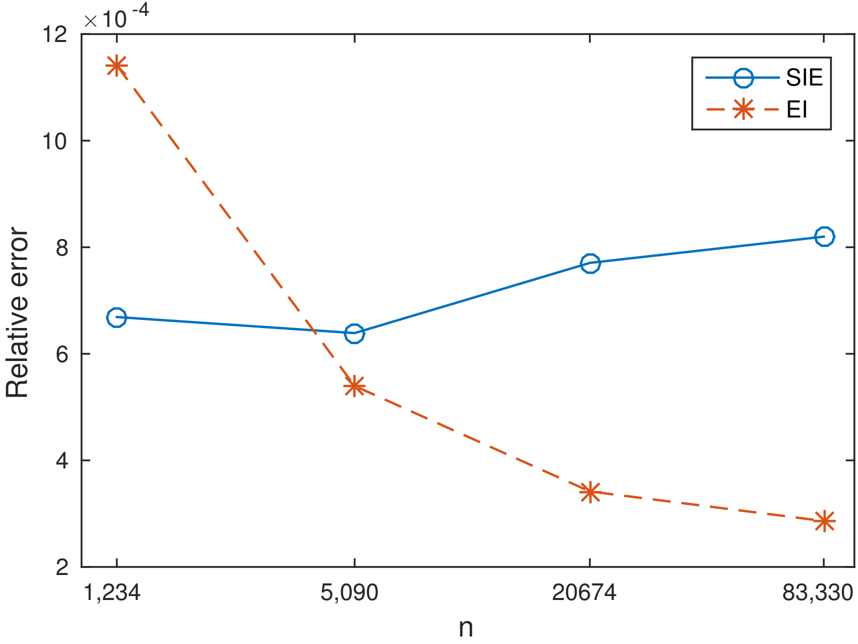


Table 3: Example 8, Comparison of ISIAP, SIAP, and AP.

n	Algorithm	CPU time(sec)	Relative error
1234	AP	1.62	1.1e–03
	SIAP	1.29	1.1e–03
	ISIAP	1.01	1.1e–03
5090	AP	7.50	5.4e–04
	SIAP	6.95	5.4e–04
	ISIAP	4.71	5.4e–04
20674	AP	42.44	3.4e–04
	SIAP	46.08	3.4e–04
	ISIAP	26.91	3.4e–04

Figure 4: Example 8, The relative error of SIE and EI of $\Delta t = 10^{-2}$.



in equation (2) with $F(y) = Ly + n(t)$. The solution at $t = 5$ is computed with the exponential integrator of $s = 1$ and $r = 2$, and $L_i = L$. The AP, SIAP, and ISIAP are used to compute the ϕ-functions in equation (8). We compare the CPU times at $\alpha = 1/50,\ 1/100,\ 1/500$. Table 4 details the results. For the SIAP and the ISIAP, $\gamma = 0.1$, and for the ISIAP, $\delta = 10^{-2}$ and $m^{\max} = 100$. The residual tolerance is 10^{-8} for ϕ-functions. The time step $\Delta t = 5 \times 10^{-2}$ is used for all the algorithms. The SIAP is the fastest. Figure 5 shows the solutions computed with ISIAP, $n = 20898$ and $\alpha = 1/500$. We see the exactness of the computational results here.

Figure 5: Example 9, $\alpha = 1/500$, $n = 20898$: Computational solution.

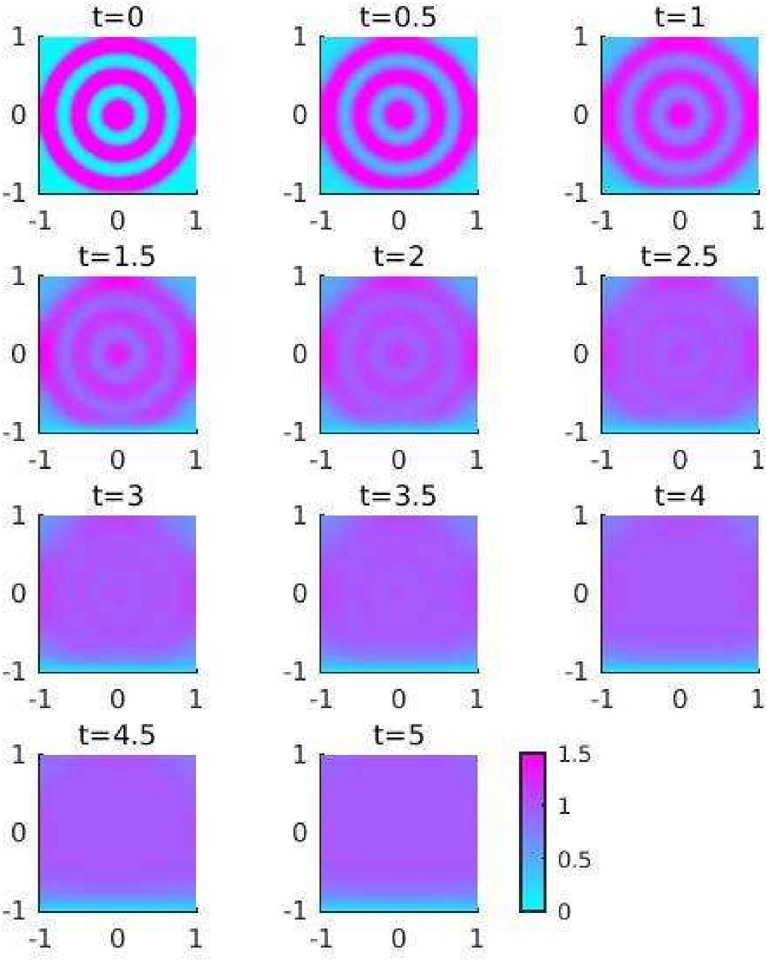


Table 4: Example 9, Comparison of the ISIAP, SIAP, and AP.

$n = 5266$			$n = 20898$		
α	Algorithm	CPU (sec)	α	Algorithm	CPU (sec)
1/50	AP	11.75	1/50	AP	61.92
	SIAP	10.69		SIAP	65.74
	ISIAP	6.61		ISIAP	34.38
1/100	AP	8.89	1/100	AP	44.46
	SIAP	8.48		SIAP	44.77
	ISIAP	5.31		ISIAP	26.55
1/500	AP	6.16	1/500	AP	25.23
	SIAP	5.02		SIAP	21.78
	ISIAP	3.90		ISIAP	16.38

5 Conclusion

In this article, the ISIAP method was proposed to compute ϕ-functions in the exponential integrator. The ISIAP solves linear equations that appear in each Arnoldi step efficiently while guaranteeing that the generalized residual remains lower than the arbitrary tolerance. It was shown that the exactness needed for solving a linear equation decreased as the Arnoldi progressed. Because the computational cost of each Arnoldi step decreased, it was possible to compute the ϕ-function faster than when using the SIAP. Moreover, it was shown that the stopping criterion for the convergence of SIAP was also valid for the convergence of the ISIAP. In the future, it will be interesting to extend the ISIAP to the rational Krylov method with more than one pole.

References

[1] Benzi, M. and Boito, P., Decay properties for functions of matrices over C*-algebras. *Linear Algebra and its Applications*, 456(1): 174–198, 2014.

doi:10.1016/j.laa.2013.11.027 E11

[2] Butcher, J. C. *Numerical methods for ordinary differential equations, second edition.* John Wiley & Sons, Chichester, England, 2008. E4

[3] Carathéodory, C., Über die gegenseitige Beziehung der Ränder bei der konformen Abbildung des Inneren einer Jordanschen Kurve auf einen Kreis. *Mathematische Annalen*, 73(2): 305–320, 1913. http://gdz.sub.uni-goettingen.de/pdfcache/PPN235181684_0073/PPN235181684_0073___LOG_0029.pdf E11

[4] Carra, E. J., Turner, I. W. and Perré, P., A variable-stepsize Jacobian-free exponential integrator for simulating transport in heterogeneous porous media: Application to wood drying. *Journal of Computational Physics*, 233: 66–82, 2013. doi:10.1016/j.jcp.2012.07.024 E4, E19

[5] Gang, W., Feng, T. and Yimin, W., An inexact shift-and-invert Arnoldi algorithm for Toeplitz matrix exponential. *Numerical Linear Algebra with Applications*, 22(4): 777–792, 2015. doi:10.1002/nla.1992 E4, E13

[6] Gallopoulos, E. and Saad, Y., Efficient solution of parabolic equations by Krylov approximation methods. *SIAM Journal on Scientific Statistics*, 13(5):1236–1264, 1992. doi:10.1137/0913071 E4

[7] Göckler, T., Rational Krylov subspace methods for ϕ-functions in exponential integrators. em Karlsruher Instituts für Technologie, 2014, Ph.D. thesis. http://d-nb.info/1060425408/34 E14

[8] Güttel, S., Rational Krylov methods for operator functions. em Technischen Universität Bergakademie Freiberg, 2010, Ph.D. thesis. http://www.qucosa.de/fileadmin/data/qucosa/documents/2764/diss_guettel.pdf E14

[9] Hashimoto, Y. and Nodera, T., Inexact shift-invert Arnoldi method for linear evolution equations (Japanese). *IPSJ Journal*, 57(10): 2250–2259, 2016. https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_

main&active_action=repository_view_main_item_detail&item_id=175055&item_no=1&page_id=13&block_id=8 E14

[10] Higham, N. J., The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4): 1179–1193, 2005. doi:10.1137/04061101X E14

[11] Hochbruck, M., A short course on exponential integrators. *Series in Contemporary Applied Mathematics*, 17: 29–49, 2015. http://www.siam.org/students/g2s3/2013/lecturers/Hochbruck/Summary_Hochbruck.pdf E4

[12] Hochbruck, M. and Ostermann, A., Exponential Runge–Kutta methods for parabolic problems. *Applied Numerical Mathematics*, 53(2–4): 323–339, 2005. doi:10.1016/j.apnum.2004.08.005 E4

[13] Hochbruck, M. and Ostermann, A., Exponential integrators. *Acta Numerica*, 19:209–286, 2010. doi:10.1017/S0962492910000048 E4, E6

[14] Hochbruck, M. and Lubich, C., On Krylov subspace approximations to the matrix exponential Operator. *SIAM Journal on Numerical Analysis*, 34(5): 1911–1925, 1997. doi:10.1137/S0036142995280572 E4

[15] Hochbruck, M., Lubich, C. and Selhofer, H., Exponential integrators for large systems of differential equations. *SIAM Journal on Scientific Computing*, 19(5):1552–1574, 1997. doi:10.1137/S1064827595295337 E10

[16] Hongqing, Z., Huazhong, S. and Meiyu, D., Numerical solutions of two-dimensional Burgers' equations by discrete Adomian decomposition method. *Computers & Mathematics with Applications*, 60(3): 840–848, 2010. doi:10.1016/j.camwa.2010.05.03 E2, E18

[17] Kamel, A. K., Numerical study of Fisher's reaction-diffusion equation by the Sinc collocation method. *Journal of Computational and Applied Mathematics*, 137(2): 245–255, 2001. doi:10.1016/S0377-0427(01)00356-9 E3

[18] Lee, S., Pang, H. and Sun, H., Shift-invert Arnoldi approximation to the Toeplitz matrix exponential. *SIAM Journal on Scientific Computing*, 32(2): 774–792, 2010. doi:10.1137/090758064 E4

[19] Moret, I. and Novati, P., RD-rational approximations of the matrix exponential. *BIT Numerical Mathematics*, 44(3): 595–615, 2004. doi:10.1023/B:BITN.0000046805.27551.3b E4

[20] Moler, C. and Van Loan, C. F., Nineteen dubious ways to compute the exponential of a matrix, Twenty-Five Years Later. *SIAM Review*, 45(1): 3–49, 2003. doi:10.1137/S00361445024180 E4

[21] Novati, P., Using the restricted-denominator rational Arnoldi method for exponential Integrators. *SIAM Journal on Matrix Analysis and Applications*, 32(4): 1537–1558, 2011. doi:10.1137/100814202 E4, E8

[22] Svoboda, Z., The convective-diffusion equation and its use in building physics. *International Journal on Architectural Science*, 1(2): 68–79, 2000. http://www.bse.polyu.edu.hk/researchCentre/Fire_Engineering/summary_of_output/journal/IJAS/V1/p.68-79.pdf E2

[23] Van der Vorst, H. A., Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2): 631–644, 1992. doi:10.1137/0913035 E15

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.
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`