Research Report

KSTS/RR-16/002
May 6, 2016

Inexact shift-invert Arnoldi method for evolution equations

by

Yuka Hashimoto
Takashi Nodera

Yuka Hashimoto
School of Fundamental Science and Technology
Keio University

Takashi Nodera
Department of Mathematics
Keio University

Department of Mathematics
Faculty of Science and Technology
Keio University

3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522 Japan

KSTS/RR-16/002
May 6, 2016

Inexact Shift-invert Arnoldi Method for Evolution Equations

Yuka Hashimoto [*] Takashi Nodera [†]

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 needs 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. The results of pertinent numerical experiments have been documented to evaluate the effectiveness of this proposed algorithm.

Key Words. Shift-invert Arnoldi, ϕ-function, exponential integrator
AMS(MOS) subject classifications. 65F60, 65M22

Contents

[*] School of Fundamental Science and Technology, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, 223-8522, Japan. Email: yukahashimoto @keio.jp

[†] Department of Mathematics, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, 223-8522, Japan. Email: nodera@math.keio.ac.jp

1 Introduction

1.1 Background

Evolution equations are used in various fields, for example, the heat equation in building physics [18], the Burgers equation in fluid mechanics [12], and the reaction diffusion equation in chemistry [13]. In this paper, the following typical initial boundary value problems defined in $[0,T] \times \overline{\Omega}$ are explored:

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

where $\Omega \subseteq \mathbb{R}^d$ is an open set, $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$, $u \in \mathcal{V} \subseteq L^2([0,T] \times \overline{\Omega})$, n_b is a unit normal vector, $\mathcal{D}$ is a differential operator on $\mathcal{V}$, and ξ, η, τ_1, τ_2 are known functions. If we discretize the equation in terms of space using a finite element method, the following differential algebraic equation can be derived:

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

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

In the simplest case, such as in the heat equation, F can be represented as $F(y) = Ly + c$, where $L \in \mathbb{R}^{n\times n}$ and $c \in \mathbb{R}^n$. Both of them are constants. In this case, the solution of equation (2) is as follows, if M and L are invertible:

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

where $\phi_0(z) := e^z$. The solution can be obtained through computing the matrix exponential once. Otherwise, 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], etc. Recently, the exponential integrator [4, 7–9] has been the popular method for solving this problem. This method is more suitable for stiff problems versus the explicit and implicit Euler methods [9, 10]. In general, at each step, $\phi_k(\Delta t M^{-1}L)$ is required, where:

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

and L is the part which is regarded as "linear" in every time step, such as the Jacobian matrix, and Δt is the step size of time.

Various methods for computing the matrix exponential and ϕ functions are introduced [5, 6, 10, 14–17]. 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 [10] obtained error bounds for ϕ_0 and ϕ_1. This means AP requires a number of iterations if $||tM^{-1}L||$ is large. In order to deal with this difficulty, the Shift-invert Arnoldi method for ϕ-function (SIAP) was proposed [15, 17]. According to Novati [17], the SIAP converges independently of $||tM^{-1}L||$. Moreover, the SIAP is suited to problems like equation (2) which is explored in this paper. With this in mind, a new method for computing ϕ-functions based on SIAP, called the Inexact Shift-invert Arnoldi method for ϕ-function (ISIAP), will be explored.

In this paper, the exponential integrator and SIAP are introduced in Section 2, and the ISIAP is proposed in Section 3, where the theoretical aspect of this method is discussed. Numerical results are given in Section 4 to show the effectiveness 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 $\kappa(A)$. e_j represents the j-th column of identity matrix I. Moreover, let $\mathbb{C}^- := \{z \in \mathbb{C};\ \Re(z) < 0\}$, and $W(A) := \{u^*Au;\ u \in \mathbb{C}^n,\ ||u|| = 1\}$, the numerical range of $n \times n$ matrix A.

2 Numerical methods for evolution equations

2.1 Exponential integrator

At the i-th step, the exponential integrator rearranges F as follows:

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

where L_i is the pseudo linear part of the i-th step. For example, $L_i = \frac{\partial}{\partial y}F(y(t_0))$ for the Exponential Runge-Kutta method, $L_i = \frac{\partial}{\partial y}F(y(t_{i-1}))$ for the Exponential Rosenbrock method, 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 follows:

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

where $t_i = i\Delta t$ $(i = 0, \cdots, N)$, $t_N = T$, and y_i is the approximation of $y(t_i)$ at the i-th step. The following approximation scheme of the one-step method is obtained: for $1 \le i \le s$,

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

where c_k are scaler coefficients, and a_{kl}, b_k are coefficients which consist of ϕ-functions with the appropriate order condition.

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

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

For a larger s, various ways of choosing a_{kl}, b_k, and c_k have been suggested. Please see Hochbruck etc. [9] and the references there for more detail.

The multi-step method was also explored. The approximation scheme of the multi-step method is as follows:

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

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

$$\begin{aligned} &\nabla^0 N_i := N_i, \quad \nabla^{k+1} N_i := \nabla^k N_i - \nabla^k N_{i-1}, \\ &\gamma_0(z) = \phi_1(z), \quad z\gamma_k(z) + 1 = \sum_{k=0}^{r-1} \frac{1}{r-k} \gamma_k(z). \end{aligned}$$

The approximation of the simplest case of $r = 1$, becomes equation (7).

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 that the ϕ-functions appear in equations (3), (6), and (8). In the case of equation (3), v is replaced with $M(v + L^{-1}c)$.

Let $\beta = ||M^{-1}v||$. Then, compute the m step Arnoldi process for $(I - \gamma M^{-1}L)^{-1} = (M - \gamma L)^{-1}M$ with the initial vector $v_1 = M^{-1}v/\beta$ from which the following relation is derived:

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

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. If H_m is invertible, $\phi_k(tM^{-1}L)M^{-1}v$ is approximated as follows:

$$\phi_k(tM^{-1}L)M^{-1}v \approx \beta V_m \psi_k(H_m^{-1}) e_1 =: u_m(t),$$

where $\psi_k(z) := \phi_k(t(1-z)/\gamma)$.

The following proposition regarding the error bound of this approximation is proved [17, Proposition 12].

Proposition 2.1 *Let* $0 \leq \theta < 0.48124$, *and* $S_\theta := \{z \in \mathbb{C}^-; \ |\arg(-z)| \leq \theta\}$. *If* $W(M^{-1}L) \subseteq S_\theta$ *and* $t/\gamma = (m+k)/\cos\theta$, *then the following error bound is held:*

$$||\phi_k(tM^{-1}L)v - u_m(t)|| \leq 11\, C\rho(\theta)^m, \tag{10}$$

where

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

and $1 \leq C \leq 11.08$.

Note that the right hand side of inequality (10) only depends on θ. Thus, proposition 2.1 implies that if γ is chosen accurately, the convergence will not depend on $||tM^{-1}L||$.

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, the assumption is that $W((I-\gamma M^{-1}L)^{-1}) \subseteq \mathbb{C}^-$. Computing this with AP requires a product of $M^{-1}L$ and a vector v_m at the m-th step, so it is necessary to solve one linear equation: $Mx_m = Lv_m$ for x_m. The computation with the SIAP also required solving one 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 $||tM^{-1}L||$, 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.

Let $F_m := [f_1 \ \cdots \ f_m]$, where $f_m := x_m - \tilde{x}_m$ is the error vector for solving the linear equation at the m-th step, and let $R_m := [r_{sys,1} \ \cdots \ r_{sys,m}]$, where $r_{sys,m} := Mv_m - (M - \gamma L)\tilde{x}_m$ is the residual vector for solving the linear equation. The following relation is derived by computing the m step Arnoldi process for $(M-\gamma L)^{-1}M$ with the same initial vector as Section 2.2:

$$\begin{aligned}(M-\gamma L)^{-1}MV_m - F_m &= V_mH_m + h_{m+1,m}v_{m+1}e_m^T, && (11)\\ MV_m - R_m &= (M-\gamma L)V_mH_m \\ &\quad + h_{m+1,m}(M-\gamma L)v_{m+1}e_m^T, && (12)\end{aligned}$$

where V_m is the $n\times m$ matrix whose columns are orthonormal, and H_m is an $m\times m$ upper Hessenberg matrix. Note that the V_m and H_m in equation (11) and (12) are different matrices from equation (9). If H_m is invertible, $\phi_k(tM^{-1}L)M^{-1}v$ is approximated as follows:

$$\phi_k(tM^{-1}L)M^{-1}v \approx \beta V_m\psi_k(H_m^{-1})e_1 =: u_m(t). \qquad (13)$$

The error of this approximation E_m is represented using Cauchy's integral formula:

$$\begin{aligned}E_m &= \psi_k(M^{-1}(M-\gamma L))M^{-1}v - \beta V_m\psi_k(H_m^{-1})e_1,\\ &= \frac{1}{2\pi i}\int_\Gamma \psi_k(\lambda)\left((\lambda I - M^{-1}(M-\gamma L))^{-1}M^{-1}v - \beta V_m(\lambda I - H_m^{-1})^{-1}e_1\right)\,d\lambda,\\ &= \frac{1}{2\pi i}\int_\Gamma \psi_k(\lambda)\left((\lambda M - (M-\gamma L))^{-1}v - \beta V_m(\lambda I - H_m^{-1})^{-1}e_1\right)\,d\lambda,\\ &= \frac{1}{2\pi i}\int_\Gamma \psi_k(\lambda)e_m\,d\lambda, \qquad (14)\end{aligned}$$

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

$$\begin{aligned}r_m &= v - (\lambda M - (M-\gamma L))\beta V_m(\lambda I - H_m^{-1})^{-1}e_1,\\ &= v - \beta\lambda MV_m(\lambda I - H_m^{-1})^{-1}e_1 + \beta\left(MV_mH_m^{-1}\right.\\ &\qquad \left. + R_mH_m^{-1} - h_{m+1,m}(M-\gamma L)v_{m+1}e_m^TH_m^{-1}\right)(\lambda I - H_m^{-1})^{-1}e_1,\\ &= v - \beta MV_m(\lambda I - H_m^{-1})(\lambda I - H_m^{-1})^{-1}e_1\end{aligned}$$

$$\begin{aligned}&+ (\beta R_m H_m^{-1} - \beta h_{m+1,m}(M - \gamma L)v_{m+1}e_m^T H_m^{-1})(\lambda I - H_m^{-1})^{-1}e_1,\\ &= (\beta R_m H_m^{-1} - \beta h_{m+1,m}(M - \gamma L)v_{m+1}e_m^T H_m^{-1})(\lambda I - H_m^{-1})^{-1}e_1.\end{aligned}$$

Replacing e_m with r_m in equation (14), the generalized residual $r_{exp,m}^{real}$ [11] of approximating $\phi_k(tM^{-1}L)M^{-1}v$ is obtained:

$$\begin{aligned}r_{exp,m}^{real} &= -\beta h_{m+1,m}(M - \gamma L)v_{m+1}e_m^T H_m^{-1}\psi_k(H_m^{-1})e_1\\ &\quad + \beta R_m H_m^{-1}\psi_k(H_m^{-1})e_1. \qquad (15)\end{aligned}$$

In order to evaluate equation (15), the following proposition is used:

Proposition 3.1 *Let $f(z) := \beta z^{-1}\psi(z^{-1})$. If*

$$W(H_m) \subseteq \mathbb{C}^- \qquad (16)$$

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

$$\left|(f(H_m))_{i,j}\right| \leq K\lambda^{i-j} \quad (i \geq j). \qquad (17)$$

Proof: Because of the boundedness of $W(H_m)$ and the assumption, there is a simply connected compact Jordan region $\mathcal{F}$ which satisfies the 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 $\Phi : \bar{\mathbb{C}}\backslash\mathcal{F} \mapsto \{w \in \bar{\mathbb{C}};\ |w| > \rho\}$ which satisfies the 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 can be 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 $R_0 > \rho$ such that the Jordan region of $\Psi\left(\{w \in \bar{\mathbb{C}};\ |w| = R_0\}\right)$ 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] $\square$.

This proposition means that if condition (16) is satisfied, the entries of $f(H_m)$ decays exponentially along the diagonal. If $||r_{sys,m}|| \leq \delta$, $(\delta > 0)$, the upper bound of the first term of equation (15) can be estimated as follows:

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

Because $0 < \lambda < 1$, the first term of equation (15) becomes smaller as m becomes larger.

Remark 3.1 *If $F_m = O$, $W(H_m)$ 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 (16), a smaller δ should be chosen to minimize the error in solving the linear equation, or a smaller γ should be chosen to separate $W((I - \gamma M^{-1}L)^{-1})$ from the origin in the complex plain.*

In the second term of equation (15), the following theorem can be deduced:

Theorem 3.1 *Let* $(f(H_m))_{i,j} =: g^m_{i,j}$*, and let* $tol_{phi} > 0$ *be the convergence threshold for computing the* ϕ*-function. If*

$$||r_{sys,1}|| \leq \frac{tol_{phi}}{m_{max}||f(H_m)e_1||}, \tag{18}$$

$$||r_{sys,j}|| \leq \frac{|g^m_{1,1}|}{|g^m_{j-1,1}|}||r_{sys,1}|| \quad (2 \leq j \leq m), \tag{19}$$

then we have:

$$||R_m f(H_m)e_1|| \leq tol_{phi}.$$

Proof: Let m_{max} be the largest number of iterations. Based on the above assumptions (18), (19) and Proposition 3.1, the following upper bound is derived:

$$\begin{aligned}
||R_m f(H_m)e_1|| &\leq |g^m_{1,1}|\,||r_{sys,1}|| + |g^m_{2,1}|\,||r_{sys,2}|| + \cdots + |g^m_{m,1}|||r_{sys,m}||, \\
&\leq |g^m_{1,1}|\,||r_{sys,1}|| + |g^m_{2,1}|\frac{|g^m_{1,1}|}{|g^m_{1,1}|}||r_{sys,1}|| \\
&\quad + |g^m_{3,1}|\frac{|g^m_{1,1}|}{|g^m_{2,1}|}||r_{sys,1}|| + \cdots + |g^m_{m,1}|\frac{|g^m_{1,1}|}{|g^m_{m-1,1}|}||r_{sys,1}||, \quad (\because (19)) \\
&= |g^m_{1,1}|\,||r_{sys,1}||\left(1 + \frac{|g^m_{2,1}|}{|g^m_{1,1}|} + \frac{|g^m_{3,1}|}{|g^m_{2,1}|} + \cdots + \frac{|g^m_{m,1}|}{|g^m_{m-1,1}|}\right), \\
&\leq ||f(H_m)e_1||\,||r_{sys,1}||\,(1 + \lambda + \cdots + \lambda), \quad (\because (17)) \\
&\leq ||f(H_m)e_1||\,||r_{sys,1}|| \cdot m_{max}, \\
&\leq tol_{phi}. \quad (\because (18)) \quad \square
\end{aligned}$$

Note that the right hand side of inequality (19) becomes larger as m becomes larger because of Proposition 3.1. Thus, the theorem implies that the larger m becomes, the solution of linear equation $(M - \gamma L)x_m = Mv_m$ becomes increasingly inexact, and the computational cost decreases. However, if the linear equations are solved, satisfying inequalities (18) and (19), the second term of equation (15) is no longer an issue. In this scenario, the first term of equation (15), which we have defined as $r^{comp}_{phi,m}$, can be used as the stopping criterion for the convergence of ISIAP.

Remark 3.2 *For* ϕ_0*, the following standard residual is available [5]:*

$$r^{real}_{phi,m} = -\frac{\beta}{\gamma}h_{m+1,m}\left(e^T_m H^{-1}_m \psi_0(H^{-1}_m)e_1\right)(I + \gamma A)v_{m+1} + \frac{\beta}{\gamma}R_m H^{-1}_m \psi_0(H^{-1}_m)e_1.$$

We can apply the same discussion for this residual.

Remark 3.3 *In practical computation, the values depending on* m *in equations (18) and (19) are unavailable in advance. Thus, the following approximation for computing equation (3) is used:*

$$||f(H_m)e_1||\,||r_{sys,1}|| \approx ||\beta V^T_m M^{-1}(M - \gamma L)V_m \psi_k(H^{-1}_m)e_1||\,||r_{sys,1}||, \quad (\because (11))$$

Algorithm 1 Inexact Shift-invert Arnoldi method (ISIAP)

```
Require: $L, M \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^n$, $t \in (0, T]$, $\gamma > 0$, $\delta > 0$, $tol_{phi} > 0$, $m_{max}$
Ensure: $u_m(t)$ such that $||r^{real}_{exp,m}|| \leq tol_{phi}$
  $\beta = ||M^{-1}v||$, $v_1 = M^{-1}v/\beta$
  $tol_{sys,1} = tol_{phi}/(m_{max}||f^i_{m(i)}||)$
  for $m = 1, 2, \cdots, m_{max}$ do
    Compute $\tilde{x}$ such that $||Mv_m - (M - \gamma L)\tilde{x}|| \leq tol_{sys,m}$
    for $l = 1, 2, \cdots, m$ do
      $h_{l,m} = \tilde{x}^T v_l$
      $\tilde{x} = \tilde{x} - h_{l,m}v_k$
    end for
    $h_{m+1,m} = ||\tilde{x}||$, $v_{m+1} = \tilde{x}/h_{m+1,m}$
    $f^{i+1}_m = H^{-1}_m \psi_k(H^{-1}_m)e_1$
    $r = |h_{m+1,m}(f^{i+1}_m)_1|||(M - \gamma L)v_{m+1}||$
    $tol_{sys,m+1} = \min\{tol_{sys,1}|(f^{i+1}_m)_1|/|(f^{i+1}_m)_m|, \delta\}$
    if $r \leq tol_{phi}$ then
      $m(i + 1) = m$
      $y_{m(i+1)}(t) = V_{m(i+1)}\psi_k(H^{-1}_{m(i+1)})e_1$, break
    end if
  end for
```

$$\approx ||M^{-1}(M - \gamma L)y(t)||\,||r_{sys,1}||, \quad (\because (13))$$
$$\approx ||M^{-1}(M - \gamma L)(v + L^{-1}c)||\,||r_{sys,1}||.$$

The matrices and vectors in the m dimensional Krylov subspace are approximated with the ones in the original space. $y(t)$ with $y(0)$ are also approximated. For computing equation (6) and (8) for the exponential integrator at the $i+1$-th step, these are replaced with the ones in the largest Krylov subspace at the i-th step. Concerning inequality (19), K and λ in inequality (17) do not depend on m, so the following approximation is used:

$$|g^m_{1,1}| \approx |g^{j-1}_{1,1}|, \quad |g^m_{1,j-1}| \approx |g^{j-1}_{1,j-1}| \quad (2 \leq j \leq m).$$

In summary, we propose Algorithm 1 for ϕ-functions in equation (6) and (8), where $(f_m)_j$ is j's element of f_m. The algorithm for computing equation (3) can be obtained through replacing the second line with $tol_{sys,1} = tol_{phi}/(m_{max}||M^{-1}(M - \gamma L)(v + L^{-1}c)||)$.

4 Numerical experiments

In this section, a few typical numerical experiments were implemented 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 3.50GHz with a Ubuntu14.04LTS operation 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 [19] 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}.

Table 1: Example 1, Comparison of ISIAE, SIAE, and AE.

n	Algorithm	CPU time(sec)	Iterations	Relative error
1925	AE	0.30	106	$2.1e-09$
	SIAE	0.18	50	$6.1e-09$
	ISIAE	0.12	50	$6.1e-09$
7561	AE	1.72	183	$8.2e-09$
	SIAE	0.53	54	$1.9e-07$
	ISIAE	0.34	54	$1.9e-07$
29969	AE	15.44	339	$3.3e-08$
	SIAE	2.18	55	$1.6e-07$
	ISIAE	1.29	55	$1.6e-07$

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

$$\begin{cases} \rho c_v \dfrac{\partial u}{\partial t} = \lambda \Delta u - \nabla \cdot cu & \text{in } (0, T] \times \Omega \\ u = 300 & \text{on } \{0\} \times \Omega \\ -\lambda \dfrac{\partial u}{\partial n} = \alpha(u - 280) & \text{on } (0, T] \times \partial\Omega_1 \\ -\lambda \dfrac{\partial u}{\partial n_b} = -1 & \text{on } (0, T] \times \partial\Omega_2 \end{cases}$$

where $\partial\Omega_2 = \{0.5\} \times [-1, 1]$, $\partial\Omega_1 = \partial\Omega \setminus \partial\Omega_1$, $c = [5\ 0]$, $\rho = 1.29$, $c_v = 1000$, $\lambda = 0.025$, $\alpha = 9.3$. 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, after which the CPU time, iteration numbers and relative errors, are compared. Please refer to Table 1 for detailed results. The relative residual tolerance tol_{phi} for computing $\phi_0(tM^{-1}L)(v + L^{-1}c)$ is 10^{-8}, and $t = 300$. $r^{comp}_{phi,m}$ is replaced with $r^{comp\,\prime}_{phi,m} := r^{comp}_{phi,m}/||M^{-1}(v + L^{-1}c)||$ 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 AP needs. This is because $||tM^{-1}L||$ 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, while 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 $r^{real\ \prime}_{phi,m} := r^{real}_{phi,m}/||M^{-1}(v+L^{-1}c)||$ decreased until it reached tol_{phi}, but it stopped decreasing after this point. It means that the linear equation can be solved efficiently at each Arnoldi step. On the other hand, the computing residual $r^{comp\,\prime}_{phi,m}$ decreased even after it reached tol_{phi}. Moreover, the behavior of $r^{real\ \prime}_{phi,m}$ and $r^{comp\,\prime}_{phi,m}$ are the same before they reach tol_{phi}. Thus, $r^{comp\,\prime}_{phi,m}$ is appropriate for the stopping criterion. Table 2 shows the residual tolerance for solving linear equations at each Arnoldi step for $n = 29969$. We can 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.

Table 2: Example 1, $n = 29969$: The residual tolerance $tol_{sys,m}$ for solving linear equations at each Arnoldi step m.

m	$tol_{sys,m}$	m	$tol_{sys,m}$	m	$tol_{sys,m}$
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	$1.5e-03$

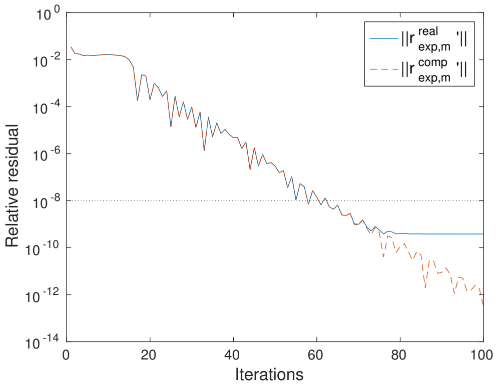

Figure 1: Example 1, $n = 29969$, Iterations vs. $||r^{real}_{phi,m}{}'||$ and $||r^{comp}_{phi,m}{}'||$.

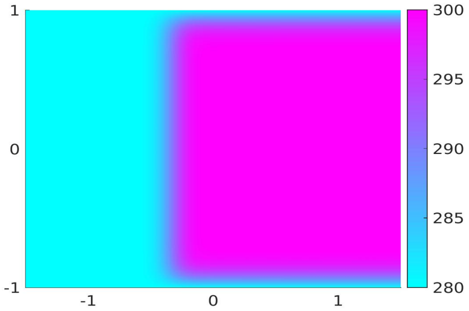

Figure 2: Example 1, $n = 29969$: Computational solution.

Example 2

The second test problem is Burgers equation in region $\Omega = (0,1) \times (0,1) \subseteq \mathbb{R}^2$:

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

where $Re = 100$, $u_{anal} = 3/4 - 1/(4 + 4e^{Re(-t-4x_1+4x_2)/32})$ and $v_{anal} = 3/4 + 1/(4 + 4e^{Re(-t-4x_1+4x_2)/32})$. The analytic solution of this problem is u_{anal} and v_{anal} [12]. 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 $s = 1$, $r = 1$ (EI), are compared. The following scheme is used for SIE:

$$B\frac{y_{i+1} - y_i}{\Delta t} = L_{i+1}y_{i+1} + n(t_i)$$
$$(B - \Delta t L_{i+1})(y_{i+1} - y_i) = F(y_i), \tag{20}$$

where $L_i = L + Q(y_{i-1})$. The linear equation (20) is solved with the BiCGStab with an ILU(0) preconditioner. The same pseudo linear part L_i is used for EI. Note that the

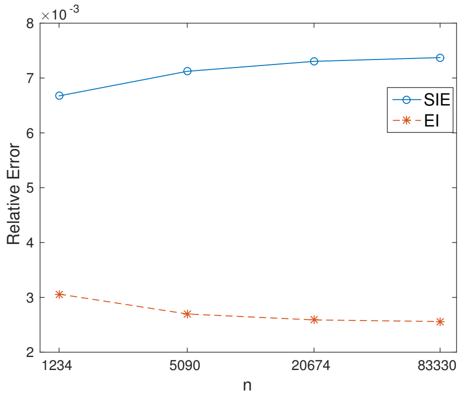

Figure 3: Example 2, The relative error of SIE and EI of $\Delta t = 10^{-1}$.

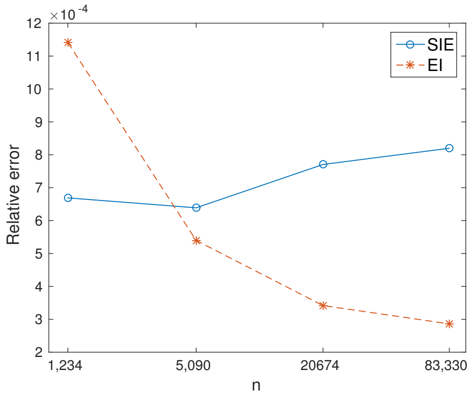

Figure 4: Example 2, The relative error of SIE and EI of $\Delta t = 10^{-2}$.

Table 3: Example 2, 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	$3.4e-04$

cost of computing the exact Jacobian matrix is prohibitively high. Because of this, 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. Figures 3–4 show the relative errors at matrix dimension $n = 1234,\ 5090,\ 20674,\ 83330$ and time step $\Delta t = 10^{-1},\ 10^{-2}$. 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. ISIAP is the fastest for all n, while the relative error is more or less the same for all algorithms.

Example 3

The next test problem explores using the reaction-diffusion Brusselator equation in region

Table 4: Example 3, Comparison of the ISIAE, SIAE, and AE.

n	5266		n	20898	
α	Algorithm	CPU time(sec)	α	Algorithm	CPU time(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

$\Omega = (-1,1) \times (-1,1) \subseteq \mathbb{R}^2$:

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

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 in $t = 0$ in Figure 5. The discretization results in equation (2) with $F(y) = Ly + n(t)$. The solution of $t = 5$ is computed with the exponential integrator of $s = 1$, $r = 2$, and $L_i = L$. The AP, SIAP, and ISIAP are used to compute the ϕ-functions in equation (8), after which the different CPU times at $\alpha = 1/50,\ 1/100,\ 1/500$ are compared. Please see Table 4 for detailed 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 $\alpha = 1/500$. The exactness of the computational results can be seen here.

5 Conclusion

In this paper, 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

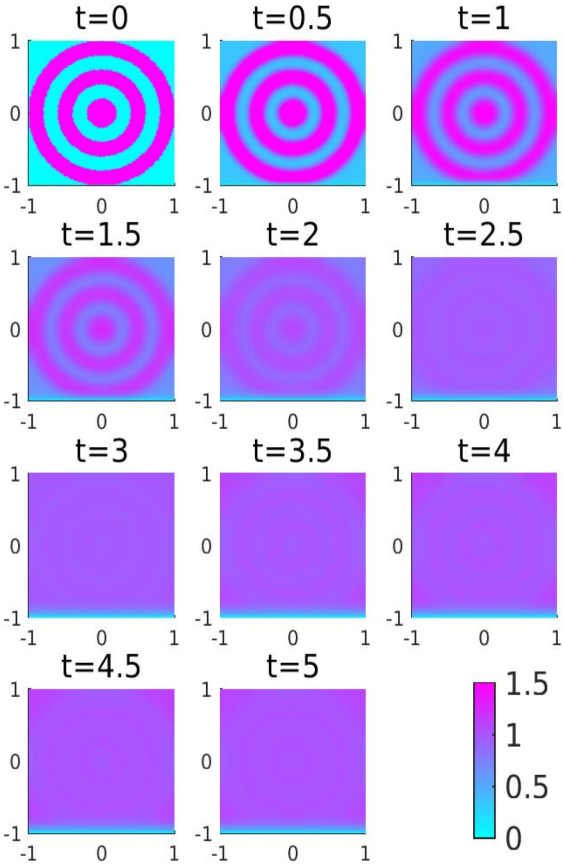

Figure 5: Example 3, $\alpha = 1/500$, $n = 20898$: Computational solution.

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. http://dx.doi.org/10.1016/j.laa.2013.11.027

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

[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.

[4] Carra, E. J., Turner, I. W. and Perré, P., A variable-stepsize Jacobian-free exponential integrator for simulating transport in heterogeneous porous media: Ap-

plication to wood drying. *Journal of Computational Physics*, 233: 66–82, 2013. http://dx.doi.org/10.1016/j.jcp.2012.07.024

[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. http://dx.doi.org/10.1002/nla.1992

[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. http://dx.doi.org/10.1137/0913071

[7] Hochbruck, M., A short course on exponential integrators. *Series in Contemporary Applied Mathematics*, 17: 29–49, 2015.

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

[9] ———, Exponential integrators. *Acta Numerica*, 19:209–286, 2010. http://dx.doi.org/10.1017/S0962492910000048

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

[11] 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. http://dx.doi.org/10.1137/S1064827595295337

[12] 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. http://dx.doi.org/10.1016/j.camwa.2010.05.03

[13] 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. http://dx.doi.org/10.1016/S0377-0427(01)00356-9

[14] 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. http://dx.doi.org/10.1137/090758064

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

[16] 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. http://dx.doi.org/10.1137/S00361445024180

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

[18] Svoboda, Z., The convective-diffusion equation and its use in building physics. *International Journal on Architectural Science*, 1(2): 68–79, 2000.

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

Department of Mathematics
Faculty of Science and Technology
Keio University

Research Report

2015

[15/001] Shiro Ishikawa,
Linguistic interpretation of quantum mechanics: Quantum Language,
KSTS/RR-15/001, January 22, 2015

[15/002] Takuji Arai, Ryoichi Suzuki,
Local risk-minimization for Lévy markets,
KSTS/RR-15/002, February 2, 2015

[15/003] Kazuma Teramoto, Takashi Nodera,
Lanczos type method for computing PageRank,
KSTS/RR-15/003, March 9, 2015

[15/004] Yoichi Matsuo, Takashi Nodera,
Block symplectic Gram-Schmidt method,
KSTS/RR-15/004, March 9, 2015

[15/005] Yuto Yokota, Takashi Nodera,
The L-BFGS method for nonlinear GMRES acceleration,
KSTS/RR-15/005, March 9, 2015

[15/006] Takatoshi Nakamura, Takashi Nodera,
The flexible incomplete LU preconditioner for large nonsymmetric linear systems,
KSTS/RR-15/006, April 13, 2015

[15/007] Takuro Kutsukake, Takashi Nodera,
The deflated flexible GMRES with an approximate inverse preconditioner,
KSTS/RR-15/007, April 15, 2015

[15/008] Dai Togashi, Takashi Nodera,
The GKB-GCV method for solving the general form of the Tikhonov regularization,
KSTS/RR-15/008, September 29, 2015

[15/009] Shiro Ishikawa,
The projection postulate in the linguistic interpretation of quantum mechanics,
KSTS/RR-15/009, November 8, 2015

2016

[16/001] Shiro Ishikawa,
Linguistic interpretation of quantum mechanics: Quantum Language [*Ver. 2*],
KSTS/RR-16/001, January 8, 2016

[16/002] Yuka Hashimoto, Takashi Nodera,
Inexact shift-invert Arnoldi method for evolution equations,
KSTS/RR-16/002, May 6, 2016