An Accurate Restarting for Shift-and-Invert Krylov Subspaces Computing Matrix Exponential Actions of Nonsymmetric Matrices

M. A. Botchev a,b,*

aKeldysh Institute of Applied Mathematics, Russian Academy of Sciences, Moscow, 125047 Russia
bMarchuk Institute of Numerical Mathematics, Russian Academy of Sciences, Moscow, 119333 Russia
*e-mail: botchev@ya.ru

Received December 24, 2020; revised December 24, 2020; accepted January 14, 2021

Abstract—An accurate residual-time (AccuRT) restarting for computing matrix exponential actions of nonsymmetric matrices by the shift-and-invert (SAI) Krylov subspace method is proposed. The proposed restarting method is an extension of the recently proposed RT (residual-time) restarting and it is designed to avoid a possible accuracy loss in the conventional RT restarting. An expensive part of the SAI Krylov method is solution of linear systems with the shifted matrix. Since the AccuRT algorithm adjusts the shift value, we discuss how the proposed restarting can be implemented with just a single LU factorization (or a preconditioner setup) of the shifted matrix. Numerical experiments demonstrate an improved accuracy and efficiency of the approach.

Keywords: shift-and-invert Krylov subspace methods, exponential time integration, Arnoldi process, Krylov subspace restarting

DOI: 10.1134/S0965542521050031

1. INTRODUCTION

Computing action of the matrix exponential on a given vector is a common task often occurring in various applications such as time integration [1], network analysis [2] or model order reduction [31]. If the matrix is large then Krylov subspace methods form an important group of methods that are well suited for this task [4]. Other methods for computing matrix exponential actions for large matrices include Chebyshev polynomials [5] and the scaling and squaring method combined with Taylor series [6]. Krylov subspace computations of the matrix exponential and other matrix functions has been an active research area, with many important developments such as rational Krylov subspace methods [7–11], restarting techniques [12–17] and challenging large-scale computational applications [9, 18–21].

Restarting techniques allow to keep the number of Krylov basis vectors (i.e., the Krylov subspace dimension) bounded while trying to preserve convergence properties of the method. Recently proposed residual-time (RT) restarting for computing matrix exponential actions seems to be an attractive technique [22]. One of its advantages is that the RT-restarted polynomial Krylov subspace method is guaranteed to converge to a required tolerance for any restart length [22]. Another nice feature of the RT restarting is that the size of the small projected problem which has to be solved in the course of iterations does not grow with each restart (as is the case for some other restarting methods, see, e.g. [23, Ch. 3]). For instance, for a restart length 10 the Krylov subspace dimension and the size of the projected problem are at most 10. In addition, the projected problem in the RT-restarted Krylov subspace methods is evaluation of the matrix exponential of a small matrix (a projection of the original large matrix). This is a simple problem which can be efficiently carried out by one of the standard linear algebra techniques [24–26]. In contrast, the projected problem in the so-called residual-based restarting [12, 27] is a system of nonautonomous ordinary differential equations (ODEs), see [12, formula (3)]. Although this ODE system is small-sized, its solution is typically more costly and requires additional care, such as a proper choice of an ODE solver and its parameters.

An important class of rational Krylov subspace methods [11] for the matrix exponential is shift-and-invert (SAI) Krylov subspace method [7, 8]. It often appears to be efficient in various applications because it requires solution of linear systems with a single shifted matrix only. One of the shortcomings of the RT
restarting proposed in [22] is that the accuracy of the SAI Krylov method combined with RT restarting can not be guaranteed. This paper proposes an extension of the RT restarting for the SAI Krylov subspace method which is designed to attain a required accuracy. The proposed technique works for nonsymmetric matrices. Moreover, we show how to implement the proposed modification efficiently, so that an LU factorization or a preconditioner setup has to be carried out only once.

The paper is organized as follows. Our restarting technique, which we call AccuRT (accurate residual–time restarting) is described in Section 2. There we first give some background information on Krylov subspace methods (Section 2.1), then we introduce and analyze AccuRT (Section 2.2) and discuss how to organize solution of the shifted systems efficiently (Section 2.3). In Section 3 numerical experiments for two test problems are presented. Finally, we draw some conclusions in Section 4.

2. ACCURT: AN ACCURATE RESIDUAL–TIME RESTARTING

Let, unless indicated otherwise, \( \| \| \) denote the 2-norm. Throughout the paper we assume that for \( A \in \mathbb{R}^{n \times n} \) holds

\[
\text{Re}(x^*Ax) \geq 0 \quad \forall x \in \mathbb{C}^n,
\]

where \( \text{Re}(z) \) denotes the real part of a complex number \( z \in \mathbb{C} \).

2.1. Krylov Subspace Methods and RT Restarting

Assume that for given matrix \( A \in \mathbb{R}^{n \times n}, v \in \mathbb{R}^n \) and \( t > 0 \), the action of the matrix exponential of \(-tA\) on the vector \( v \) has to be computed, i.e.,

\[
\text{compute } y := \exp(-tA)v.
\]

The problem is equivalent to solving initial value problem (IVP)

\[
y'(t) = -Ay(t), \quad y(0) = v,
\]

where we slightly abuse the notation by using \( t \) for both the independent variable in (3) and for time interval length in (2). Krylov subspace method for computing the matrix exponential action can be seen as a Galerkin projection of IVP (3) on the Krylov subspace

\[
\mathcal{K}_k(A,v) = \text{span}(v, Av, A^2v, \ldots, A^{k-1}v).
\]

First, an orthonormal basis of \( \mathcal{K}_k(A,v) \) is computed by the standard Arnoldi (or, if \( A = A^\top \), by Lanczos) process [24, 28–30] and stored as the columns \( v_1, \ldots, v_k \) of a matrix \( V_k = [v_1 \ldots v_k] \in \mathbb{R}^{n \times k} \), such that the following Arnoldi decomposition holds:

\[
AV_k = V_{k+1}H_k = V_kH_k + h_{k+1}v_{k+1}e_k^\top,
\]

where \( e_k = (0, \ldots, 0, 1)^\top \in \mathbb{R}^k \), \( H_k \in \mathbb{R}^{(k+1) \times k} \) is an upper Hessenberg matrix and \( H_k \in \mathbb{R}^{k \times k} \) contains the first \( k \) rows of \( H_k \). Then the Krylov subspace approximation \( y_k(t) = \exp(-tA)v \) is defined as [31–33]

\[
y_k(t) = V_ku(t),
\]

where \( u(t) : \mathbb{R} \to \mathbb{R}^k \) solves IVP with the projected matrix \( H_k = V_k^\top AV_k \)

\[
u'(t) = -H_ku(t), \quad u(0) = \beta e_1.
\]

Here \( \beta = \|v\| \) and \( e_1 = (1, 0, \ldots, 0)^\top \in \mathbb{R}^k \). Note that IVP (7) is a Galerkin projection of (3) on the Krylov subspace and that \( u(t) \) can be computed as

\[
u(t) = \exp(-tH_k)\beta e_1.
\]

If \( k \) is not too large, (8) is a preferred way to compute \( u \) which can be implemented by one of the standard dense linear algebra routines [24–26]. Computing \( \exp(-tH_k) \) is usually quicker than solving the ODE system in (7), which requires choosing a suitable ODE solver (e.g., stiff or non-stiff) and its parameters. Kry-
lov subspace method (5)–(8) is sometimes called polynomial Krylov subspace method to emphasize the fact that vectors of the subspace (4) are polynomials in $A$.

A natural way of controlling the (unknown) error of the Krylov subspace approximation (6) is to monitor the residual $r_k(t)$ of $y_k(t)$ with respect to the ODE $y' = -Ay$, namely [12, 27, 34]

$$r_k(t) = -Ay_k(t) - y_k(t).$$

The residual $r_k(t)$ is readily available in the course of the Arnoldi process computations as [12, 27, 34]

$$r_k(t) = -h_{k+1,k} e_k^T u(t) v_{k+1},$$

where we see that $r_k(t)$ is a scalar function times $v_{k+1}$. Hence, $V_k^T r_k(t) = 0$ for all $t > 0$ and (6) is indeed a Galerkin projection on $\mathcal{H}_k(A,v)$. Some results on residual convergence and its connection to the error can be found, e.g., in [22, 27].

The SAI (shift-and-invert) Krylov subspace method [7, 8] for computing (2) differs from the standard polynomial Krylov subspace method described above in that it builds the Krylov subspace for the shifted-and-inverted matrix $(I + \gamma A)^{-1}$ instead of $A$, with $\gamma > 0$ being a fixed parameter. This is done to accelerate convergence: the Arnoldi process tends to emphasize the largest eigenvalues of $(I + \gamma A)^{-1}$ corresponding to the smallest eigenvalues of $A$. The latter are important for the exponential which is a quickly decaying function (assuming real arguments). The price to pay for this accelerated convergence is the necessity to solve a linear system with the matrix $H_k$ at each Krylov step. The Arnoldi decomposition (5) for the SAI matrix $(I + \gamma A)^{-1}$, i.e.,

$$(I + \gamma A)^{-1}V_k = V_{k+1} \tilde{H}_k = V_k \tilde{H}_k + h_{k+1,k} v_{k+1} e_k^T$$

is more convenient to use in a transformed form

$$AV_k = V_k H_k - \frac{h_{k+1,k}}{\gamma} (I + \gamma A)v_{k+1} e_k^T \tilde{H}_k^{-1},$$

where the notation $\gamma$ is used to indicate that the projection is built for the SAI matrix $(I + \gamma A)^{-1}$ and $H_k$ is defined as the inverse SAI transformation

$$H_k = \frac{1}{\gamma} (\tilde{H}_k^{-1} - I).$$

Note that the matrices $V_{k+1}$ and $H_k$ here are different from those in (5). The SAI Krylov subspace and related methods are further analyzed, e.g., in [7, 8, 10].

In the SAI Krylov subspace method the residual can be easily computed as [27]

$$r_k(t) = \frac{h_{k+1,k}}{\gamma} (e_k^T \tilde{H}_k^{-1} u(t))(I + \gamma A)v_{k+1},$$

where $u(t)$ is the solution of the projected IVP (7), with $H_k$ being the back transformed SAI matrix (12).

The shift value $\gamma$ is usually chosen in accordance with the length $t$ of the time interval $[0, t]$ (see [8]). A possible often used value is $\gamma = t/10$. Hence, changing $\gamma$ means changing $t$. The usual polynomial Krylov subspace method (5)–(8) has an attractive property that it is invariant of $t$: once $V_{k+1}$ and $H_k$ are computed, they can be successfully used for any $t$ (although the quality of approximation $y_k(t) = y(t)$ does deteriorate with $t$). Unfortunately, this nice property is not fully shared by the SAI Krylov subspace method: its matrices $V_{k+1}$ and $H_k$ do depend on $\gamma$ which, in turn, depends on $t$. However, in practice, one certainly can use the computed Arnoldi matrices $V_{k+1}$ and $H_k$ for a certain range of $t$ without recomputing them.

Recently proposed RT (residual-time) restarting is based on the fact that the residual as a function of $t$, for the regular Krylov subspace method, tends to be a non-decreasing function. Hence, once a maximum number $k_{\text{max}}$ of Krylov steps are done (so that storing and working with more Krylov basis vectors is too expensive), we can find a time subinterval $[0, \delta]$, $\delta < t$, where the residual norm is already sufficiently small. We can then restart the Krylov method by setting $v := y_{k_{\text{max}}} (\delta)$, decreasing the time interval $t := t - \delta$. 

\[ \text{end of page} \]
AN ACCURATE RESTARTING

and performing the next \( k_{\text{max}} \) Krylov steps solving problem (2) with the modified \( \nu \) and \( t \). The RT restarting procedure is sketched in Fig. 1.

2.2. AccuRT Ideas and Algorithm

The residual as a function of \( t \) in the SAI Krylov subspace method exhibits a much more irregular behavior than in the polynomial Krylov method (5)--(8) (see Fig. 2). If the RT restarting is applied for the SAI Krylov subspace method then it can happen that \( \delta \) such that \( \| r_k(s) \| \) is below a certain tolerance for \( s \in [0, \delta] \) can not be found or is too small to be used in practice for an efficient restarting. Of course, we could also restart by setting \( \delta \) to any as large as possible point \( s \) where \( \| r_k(s) \| \) is small enough (cf. Fig. 2).

However, there is no guarantee that \( \min_{s \in [0, t]} \| r_k(s) \| \) is within the required tolerance and, if this is the case, the restarting with any \( \delta \in [0, t] \) inevitably leads to an accuracy loss.

Here we propose an approach to fix this failure of the RT restarting in the SAI Krylov subspace method. The approach is based on the following two simple observations.

(1) Since \( \gamma \) is usually chosen proportionally to \( t \), taking a smaller shift value \( \gamma \) effectively means a shorter time interval \([0, \delta]\). For nonsymmetric matrices \( A \) the SAI Krylov residual \( r_k(s) \) tends to become smaller in norm with smaller \( \gamma \) on some time subinterval \( s \in [0, \delta] \), \( 0 < \delta < t \) (see Figs. 2, 3).

(2) As already noted, to change \( \gamma \) in the SAI Krylov subspace method we have to carry out the whole Arnoldi process anew. However, once a shifted linear system with the matrix \( I + \gamma A \) is solved for a certain shift \( \gamma \), a part of the spent computational costs can be re-used for solving the shifted linear systems with a smaller shift \( \tilde{\gamma} \leq \gamma \). In particular, if a (sparse) LU factorization is computed for a certain shift \( \gamma \), it can be successfully used as a preconditioner for solving shifted systems with \( I + \tilde{\gamma} A \), \( \tilde{\gamma} \leq \gamma \) (see Proposition 2).

Based on these observations we propose to organize an improved RT restarting for the SAI Krylov subspace method as follows. Assume we can carry out at most \( k_{\text{max}} \) steps of the Arnoldi or Lanczos process, since storing (or working with) more than \( k_{\text{max}} \) Krylov basis vectors is too expensive. We perform \( k = 1, 2, \ldots, k_{\text{max}} \) steps checking at each step the residual norm \( \| r_k(t) \| \), cf. (13). If the residual norm turns out to be smaller than the required tolerance, we stop. Otherwise, after performing step \( k = k_{\text{max}} \), we analyze the function \( \| r_k(s) \| \) for \( s \in [0, t] \). If no point \( s \) can be found where \( \| r_k(s) \| \) is small enough, we decrease \( \gamma \) by a factor of two and repeat \( k = 1, 2, \ldots, k_{\text{max}} \) steps of the method. The restarting procedure (which is presented in detail in Fig. 4) can be repeated until the residual norm is small enough at the end of the given time interval.

If a (sparse) LU factorization is prohibitively expensive, a preconditioned solver can be used to solve the shifted linear system. In the algorithm presented in Figure 4 we then replace the LU factorization step

---

**Fig. 1.** A sketch of the RT restarting procedure, adopted from [22].
by a setup of the preconditioner. The computed preconditioner can then be used for all the shift values, i.e., it suffices to set up the preconditioner once.

Note that the observed behavior of the SAI Krylov residual for decreasing \( \gamma \) does not hold for symmetric \( A \). This is reflected by a rather sharp estimate \([8, \text{Lemma 3.1}]\) (in the estimate there, take \( \mu = 0 \) and set \( \gamma \) proportional to \( t \)).

The following lemma and proposition show that the residual \( r_k(s) \) of the SAI Krylov subspace method is bounded in norm as a function of time. The way it is bounded depends on \( \gamma \).

\textbf{Lemma 1.} Let \( A \in \mathbb{R}^{n \times n} \) be a matrix for which relation (1) holds and let \( H_k \) be the matrix obtained in the SAI Krylov subspace method (see (11), (12)). Then there exists a constant \( \omega_k \geq 0 \) such that

\[
\left\| \exp(-tH_k) \right\| \leq e^{-\omega_k}.
\]
% Given: $A \in \mathbb{R}^{n \times n}, v \in \mathbb{R}^n, t > 0, k_{\text{max}}$ and tol > 0
convergence := false
$\gamma_{\text{changed}} := \text{false}$
carry out LU factorization $LU := I + \gamma A$
while (not(convergence) and $t > 0$)
\[ \beta := \|v\|, v_1 := v/\beta \]
for $k = 1, \ldots, k_{\text{max}}$
if $\gamma_{\text{changed}}$
solve $(I + \gamma A)w = v_k$ iterative,
preconditioned by $LU$
else
slove $(I + \gamma A)w = v_k$ by LU factorization
end
for $i = 1, \ldots, k$
\[ h_{i,k} := w^t v_i, w := w - h_{i,k} v_i \]
end
\[ h_{k+1,k} := \|w\| \]
\[ H_k := \frac{1}{t} (H_k^{-1} - I) \]
compute $u(s_j), \|r_k(s_j)\|, s_j = jt/3, j = 1, 2, 3$
resnorm := max $\|r_k(s_j)\|$ if resnorm $\leq$ tol and $k > 1$
convergence := true
break loop for $k = \ldots$
elseif $k = k_{\text{max}}$
%—restart at step $k_{\text{max}}$
compute $\|r_k(s_j)\|, s_j = jt/500, j = 1, \ldots, 500$
$r_{\text{min}} := \min_j \|r_k(s_j)\|$
if $r_{\text{min}} > \text{tol}$
\[ \delta := 0 \]
\[ \gamma := \gamma/2 \]
\[ \gamma_{\text{changed}} := \text{true} \]
else
\[ \delta := \max(s_j | \|r_k(s_j)\| \leq \text{tol}) \]
end
\[ u := \exp(-\delta H_k) e_1, v := V_k(\beta u) \]
\[ t := t - \delta \]
end
\[ v_{k+1} := w/h_{k+1,j} \]
end
end
\[ y_k := V_k(\beta u(s_j)) \]

Fig. 4. Description of the AccuRT restarting algorithm. The algorithm computes SAI Krylov subspace approximation $y_k(t) = \exp(-tA)v$ such that for its residual $r_k(t)$ holds $\|r_k(t)\| \leq \text{tol}$ for $s = t/3, s = 2t/3$ and $s = t$.

Proof. Let $\omega = \min_{x \in \mathbb{C}^n \setminus \{0\}} \Re(x^* Ax)$. It is well known (see, e.g., [35, Theorem 2.4]) that
\[ \Re(x^* Ax) \geq \omega \quad \forall x \in \mathbb{C}^n \iff \|\exp(-tA)\| \leq e^{-\omega t}. \]
According to (1), these two equivalent relations hold with $\omega \geq 0$. Furthermore, these conditions are equivalent to [35, Theorem 2.13]
\[ \| (I + \gamma A)^{-1} \| \leq \frac{1}{1 + \gamma \omega}, \]
which holds for all $\gamma > 0$ and all $\omega \in \mathbb{R}$ provided that $1 + \gamma \omega > 0$. Let $\gamma > 0$, as is the case in the SAI Krylov subspace method. Define
\[ \omega_k := \frac{1}{\gamma} (\|H_k\|^{-1} - 1), \]
so that \( \| \tilde{H}_k \| = 1/(1 + \gamma \omega_k) \). We have

\[
\frac{1}{1 + \gamma \omega_k} = \| \tilde{H}_k \| = \| V_k^T (I + \gamma A)^{-1} V_k \| \leq \| (I + \gamma A)^{-1} \| \leq \frac{1}{1 + \gamma \omega},
\]

from which it follows that \( 0 \leq \omega \leq \omega_k \). Since \( \tilde{H}_k = (I + \gamma H_k)^{-1} \) (cf. (12)), we obtain, again using \([35, Theorem 2.13]\),

\[
(I + \gamma H_k)^{-1} = \frac{1}{1 + \gamma \omega_k} \iff \text{Re}(x^* H_k x) \geq \omega_k \quad \forall x \in \mathbb{C}^k,
\]

which is equivalent to inequality (14) we prove.

Define function \( \varphi(z) \) as (see, e.g., \([1]\))

\[
\varphi(z) = (e^z - 1)/z.
\]

**Proposition 1.** \( A \in \mathbb{R}^{m \times n} \) be a matrix for which relation (1) holds and let \( r_k(t) \) be the residual of the SAI Krylov subspace method defined by (13), applied to solve problem (2). Then it holds for all \( t \geq 0 \)

\[
r_k(t) = \beta_k(t) w_{k+1}, \quad \beta_k(t) = \frac{\beta_{k+1,k}}{\gamma} e_k^T (I + \gamma H_k) u(t), \quad w_{k+1} = (I + \gamma A) w_{k+1},
\]

\[
\| \beta_k(t) \| = \| \beta_k(t) \| \| w_{k+1} \|, \quad \beta_k(t) \leq \beta \tilde{h}_{k+1,k} \left( \frac{1}{\gamma} \min \left\{ t \| (I + \gamma H_k) H_k \varphi(-t \omega_k), \| I + \gamma H_k \| (1 + e^{-\omega_k}) \right\} \right),
\]

where \( u(t) \) is defined by (8), (12), \( \omega_k \geq 0 \) is the constant introduced in (14) and \( \tilde{h}_{k+1,k} \) and \( h_{k,1} \) are the corresponding entries of the matrices \( \tilde{H}_k \in \mathbb{R}^{(k+1) \times k} \) and \( H_k \in \mathbb{R}^{k \times k} \), respectively (see (11), (12)). Here, the minimum is taken among the two elements of the set indicated by the braces \{\ldots\}. Note that \( \tilde{H}_k \) in the estimate above depends on \( \gamma \) (and, hence, so do \( H_k, u(t) \), and \( \omega_k \)).

**Proof.** Relation (17) is identical to (13) (cf. (12)), its proof can be found in \([27]\). Using (7), (12), we have

\[
\beta_k(t) = \frac{\beta_{k+1,k}}{\gamma} e_k^T (I + \gamma H_k) u_k(t) \quad \text{and} \quad w_{k+1} = (I + \gamma A) w_{k+1}.
\]

Furthermore, it is not difficult to check that, cf. (16),

\[
u(t) - u(0) = (\exp(-t H_k) - I) u(0) = -t H_k \varphi(-t H_k) u(0).
\]

Therefore, with \( u(0) = \beta e_k \) and \( H_k^{-1} = I + \gamma H_k \), we can estimate

\[
\| (I + \gamma H_k) u(t) - u(0) \| \leq t \| (I + \gamma H_k) H_k \varphi(-t \omega_k) \| \| u(0) \| \leq \beta t \| (I + \gamma H_k) H_k \| \| \varphi(-t \omega_k) \|.
\]

Here we used an inequality

\[
\| \varphi(-t H_k) \| \leq \varphi(-t \omega_k),
\]

which holds due to the property (14), (15), see, e.g., \([1, proof of Lemma 2.4]\). The estimate (19) is especially useful for small \( t \). An alternative estimate, which may be sharper for large \( t \), is

\[
\| (I + \gamma H_k) u(t) - u(0) \| \leq \| (I + \gamma H_k) \exp(-t H_k) - I \| \| u(0) \| \leq \beta \| I + \gamma H_k \| (1 + \| \exp(-t H_k) \|)
\]

\[
\leq \beta \| I + \gamma H_k \| (1 + e^{-\omega_k}),
\]

where the estimate (14) is used. From (19), (20) it follows that

\[
\| (I + \gamma H_k) u(t) - u(0) \| \leq \beta \min \left\{ t \| (I + \gamma H_k) H_k \| \varphi(-t \omega_k), I + \gamma H_k \| I + \gamma H_k \| (1 + e^{-\omega_k}) \right\}.
\]
We can then estimate
\[ \| \beta_k(t) \| \leq \| \beta_k(t) - \beta_k(0) \| + \| \beta_k(0) \| = \frac{\hat{h}_{k+1,k}}{\gamma} \| e_t(I + \gamma H_k)(u(t) - u(0)) \| + \beta \hat{h}_{k+1,k} \| h_{k,1} \| \]
\[ \leq \frac{\hat{h}_{k+1,k}}{\gamma} \| (I + \gamma H_k)(u(t) - u(0)) \| + \beta \hat{h}_{k+1,k} \| h_{k,1} \| \]
\[ \leq \frac{\hat{h}_{k+1,k}}{\gamma} \beta \min \{ t \| (I + \gamma H_k)H_k \| \phi(-t \omega_k), \| (I + \gamma H_k)H_k \| (1 + e^{-r_0 k}) \} + \beta \hat{h}_{k+1,k} \| h_{k,1} \| \]
\[ = \beta \hat{h}_{k+1,k} \left( \frac{1}{\gamma} \min \{ t \| (I + \gamma H_k)H_k \| \phi(-t \omega_k), \| (I + \gamma H_k)H_k \| (1 + e^{-r_0 k}) \} + \| h_{k,1} \| \right), \]
which proves (18).

We note that the estimate (18) is, unfortunately, far from sharp to reflect the noticed dependence of \( \| \beta_k(t) \| \) on \( \gamma \) (cf. Figs. 2 and 3). However, the following remark should be made.

**Remark 1.** Numerical experiments show that the value \( \| h_{k,1} \| \) (recall \( \| \beta_k(0) \| = \beta \hat{h}_{k+1,k} \| h_{k,1} \| \)) appearing in (18) is usually small, typically many orders of magnitude smaller than the other term \( \frac{1}{\gamma} \min \{ \ldots \} \) appearing in the right hand side of (18). If \( \| h_{k,1} \| = 0 \) then \( \| \beta_k(0) \| = 0 \), so that (18) formally shows that for any \( k \) and any tolerance \( \varepsilon > 0 \) a time interval \([0, \delta]\) can be found such that \( \| \beta_k(s) \| \leq \varepsilon \) for \( s \in [0, \delta] \). In this case the time interval can surely be reduced (see Fig. 1), so that both the RT and AccuRT restarting strategies are guaranteed to converge for any restart length. Of course, such a \( \delta \) can still be too small to be used in practice, so that adjusting \( \gamma \), as is done in AccuRT, may be necessary to make the restarting practical.

### 2.3. Solving the Shifted Linear Systems

We now show that an LU factorization computed for the shifted matrix \( I + \gamma A \) can be successfully used as a preconditioner for the shifted matrix \( I + \gamma A \) with an adjusted shift \( \gamma \) such that \( 0 < \gamma \leq \gamma \). More precisely, the shifted linear system
\[ Ax = b, \quad A = I + \gamma A, \]
is preconditioned as
\[ M^{-1}Ax = M^{-1}b, \quad M = I + \gamma A. \tag{21} \]
It is then not difficult to show (see Proposition 2 below) that even simple Richardson iteration for the preconditioned system (21), namely,
\[ x_{m+1} = \tilde{G}x_m + M^{-1}b, \quad \tilde{G} = I - M^{-1}A, \tag{22} \]
converge unconditionally, i.e., for the spectral radius \( \rho(\tilde{G}) \) of \( \tilde{G} \) holds \( \rho(\tilde{G}) < 1 \). Hence, the eigenvalues of the preconditioned matrix \( M^{-1}A \) are located on the complex plane inside the unit circle centered at point \( 1 + \gamma i \), \( i^2 = -1 \). This means that any other modern Krylov subspace method such as GMRES, BiCGSTAB, QMR and other [29, 30, 36] should successfully converge for the preconditioned linear system (21).

On the other hand, the smaller the shift value \( \gamma \), the better the shifted matrix \( I + \gamma A \) is conditioned. Hence, for a small \( \gamma \) it may turn out that an unpreconditioned iterative method converges fast enough. Therefore, in Proposition 2 we give a sufficient condition which guarantees that the preconditioned Richardson method converges faster than unpreconditioned one.

**Proposition 2.** Let \( A \in \mathbb{R}^{m \times n} \) be a matrix for which relation (1) holds, \( 0 < \gamma \leq \gamma \) and let a linear system with the matrix \( I + \gamma A \) be solved iteratively. Then Richardson iteration (22) with the preconditioner matrix \( M = I + \gamma A \) converges.
Furthermore, assume unpreconditioned Richardson iteration converges, too. Then Richardson iteration (22) with the preconditioner matrix \( M = I + \gamma A \) converges faster than unpreconditioned Richardson iteration provided that

\[
\frac{1}{1 + \gamma \rho(A)} < \frac{\gamma}{\gamma},
\]

where \( \rho(A) \) is the spectral radius of the matrix \( A \).

**Proof.** Let \( \lambda \) be an eigenvalue of \( A \). The eigenvalues of the preconditioned matrix \( (I + \gamma A)^{-1}(I + \gamma A) \) read

\[
1 + \frac{\gamma \lambda}{1 + \gamma \lambda} = 1 - \left(1 - \frac{\gamma}{\gamma}\right)\frac{\lambda}{1 + \gamma \lambda}.
\]

The preconditioned Richardson iteration converge if and only if all the eigenvalues of the iteration matrix \( \tilde{G} = I - (I + \gamma A)^{-1}(I + \gamma A) \) are smaller in absolute value than 1, i.e.,

\[
\left|1 - \frac{\gamma}{\gamma}\right|\frac{\lambda}{1 + \gamma \lambda} < 1.
\]

The left-hand side of this inequality can be bounded as

\[
\left|1 - \frac{\gamma}{\gamma}\right|\frac{\lambda}{1 + \gamma \lambda} \leq \frac{|\lambda|}{1 + \gamma \lambda} < 1,
\]

where the last inequality holds because all the eigenvalues of \( A \) have nonnegative real parts (see (1)). Hence, the preconditioned Richardson method converges.

Furthermore note that the unpreconditioned Richardson iteration matrix \( G \) reads \( G = I - (I + \gamma A)^{-1} = -\gamma A \). The preconditioned Richardson iteration converges faster than unpreconditioned one provided that \( \rho(\tilde{G}) < \rho(G) \), i.e.,

\[
\left(1 - \frac{\gamma}{\gamma}\right)_{\lambda} \max \left|\frac{\lambda}{1 + \gamma \lambda}\right| < \max |\lambda| = \gamma \rho(A).
\]

The left-hand side here can be bounded by \( 1 - \frac{\gamma}{\gamma} \) and we see that the inequality holds as soon as

\[
\left(1 - \frac{\gamma}{\gamma}\right) < \gamma \rho(A).
\]

It is easy to check that the last inequality is equivalent to (23).

### 3. NUMERICAL EXPERIMENTS

#### 3.1. Experiment Setup and Details

We implemented the AccuRT algorithm as shown in Fig. 4 with the following modifications.

1. The approximate smallest residual norm value \( \min_{\text{se}[\mathbb{U}]} \| \tilde{e}(s) \| \) depends on the number of sample points at which the residual is evaluated. Therefore this number, denoted in Fig. 4 by \( n_{\text{steps}} \), is set according to the required tolerance \( \text{tol} \) as follows:

\[
\begin{array}{ccc}
tol & 1e-06 & 1e-07 \\
n_{\text{steps}} & 500 & 1000 & 2000 \\
\leq 1e-08 & \\
\end{array}
\]

2. If a restart value \( \delta \) (i.e., the value such that \( \| \tilde{e}(\delta) \| \leq \text{tol} \) (see Fig. 4)) stays zero after two subsequent halvings of \( \gamma \) then we set \( \gamma \) to its initial value \( 4 \gamma \) times 0.8, i.e., we set \( \gamma := 0.8 \times 4 \times \gamma \), and double the number of the sample points \( n_{\text{steps}} \). Then the algorithm proceeds as before. This means that if \( \gamma \) is initially 1, then in the algorithm it gets values 1, 0.5, 0.25, 0.8, 0.4, 0.2, 0.64, 0.32, … . As soon as a positive \( \delta \) is found, i.e., a restart is successful, \( n_{\text{steps}} \) is set to 500.
(3) As soon as $\gamma$ is changed, we adjust the time interval, on which the residual norm minimum is searched for with $n_{\text{steps}}$ samples, proportionally. For instance, every time the value of $\gamma$ is halved, we restrict the search time interval from $[0, t]$ to $[0, t/2]$. This is done to account for the fact that the residual norm $\|\mathbf{r}_3(s)\|$ is likely not to be small for $s > t/2$ (see Fig. 3). Once a restart is successful, i.e., $\delta > 0$ and the time interval is decreased (line $t := t - \delta$), we set the search interval back to $[0, t]$.

(4) The last modification is that, as an option, GMRES(10) [37] can be used as a linear solver instead of the LU factorization, not only when $\gamma$ is decreased. In this case the ILUT($\varepsilon$) preconditioner [38, Ch. 10] can be used which is computed only once and re-used for all values of $\gamma$. We use the Templates implementation of GMRES [36] available at www.netlib.org/templates/, the preconditioner is applied from the right and the GMRES(10) iterations stop as soon as the residual $\|\mathbf{r}\|$ satisfies

$$
\|\mathbf{r}\| \leq \text{tol}_{\text{gmres}}\|\mathbf{b}\|, \quad \text{with} \quad \text{tol}_{\text{gmres}} = \min\{1e^{-08},\text{tol}/10\},
$$

where tol is the required tolerance for computing the matrix exponential action. For small values of $\gamma$ it may be sensible to switch the preconditioner off (see Proposition 2).

Initial value for $\gamma$ can be provided to our AccuRT subroutine as an optional parameter. By default, if $\gamma$ is not provided by the user, it is set to $t/20$. Note that $\gamma = t/10$ is suggested in [8] as an appropriate value for moderate tolerances $\text{tol} \approx 10^{-6}$ for symmetric matrices. Setting $\gamma$ to $t/20$ appears to be a reasonable choice because, as our limited experience suggests, optimal values of $\gamma$ for nonsymmetric matrices are usually smaller than $t/10$.

In the experiments below, within the framework of the SAI Krylov subspace method, we compare the performance of the AccuRT restarting with that of the RT restarting. In our recent work [22] the RT restarting has been compared with three other restarting strategies, namely, the EXPOKIT restarting [39], the Niehoff–Hochbruck restarting [23] and the residual restarting [12, 27]. The errors reported below in this section are relative errors computed, for a numerical solution $y_k(t)$, as

$$
\frac{\|y_k(t) - y_{\text{ref}}(t)\|}{\|y_{\text{ref}}(t)\|},
$$

where $y_{\text{ref}}(t)$ is a reference solution computed by the `phiv` function of the EXPOKIT package [39] with a high tolerance. The presented numerical tests are performed in Matlab on a Linux PC with 8 CPUs Intel Xeon E5504 2.00 GHz.

### 3.2. Convection–Diffusion Problem

The matrix $A$ in this problem is obtained by a standard five point central–difference discretization of a convection–diffusion operator defined on functions $u(x, y)$, with $(x, y) \in \Omega = [0, 1] \times [0, 1]$, and $u|_{\partial\Omega} = 0$. The operator reads

$$
L[u] = -(D_1 u_x)_x - (D_2 u_y)_y + \text{Pe} \left( \frac{1}{2} (v_1 u_x + v_2 u_y) + \frac{1}{2} ((v_1 u)_x + (v_2 u)_y) \right),
$$

$$
D_1(x, y) = \begin{cases} 
10^3 & (x, y) \in [0.25, 0.75]^2, \\
1 & \text{otherwise}, 
\end{cases} \quad D_2(x, y) = \frac{1}{2} D_1(x, y),
$$

where $\text{Pe}$ is the Peclet number. Here the special way the convective terms (the first derivatives) are written in takes care that the discretized convection terms give a skew-symmetric matrix [40]. In the experiments, we use a uniform $802 \times 802$ grid and the Peclet numbers $\text{Pe} = 200$ and $\text{Pe} = 1000$. The problem size for this grid is $n = 800^2 = 640000$. For both Peclet numbers we have $\| \frac{1}{2} (A - A^T) \|_2 \approx 6000$, whereas $\| \frac{1}{2} (A - A^T) \|_2 \approx 0.5$ for $\text{Pe} = 200$ and $\| \frac{1}{2} (A - A^T) \|_2 \approx 2.5$ for $\text{Pe} = 1000$. Hence, in both cases the matrices are weakly non-symmetric. The values of the function $\sin(\pi x)\sin(\pi y)$ on the finite-difference grid are assigned to the initial vector $\nu$, which is then normalized as $\nu := \nu/\|\nu\|$. The final time is set to $t = 1$.

In this test the initial value of $\gamma$ in the AccuRT restarting is not altered from its default value $t/20$, whereas the RT restarting uses the usual value $t/10$. This does not necessarily gives an advantage to AccuRT because optimal $\gamma$ values detected by AccuRT are smaller than $t/20$ anyway.
Results for this test problem are presented in Table 1. As can be seen in the first two lines of the table, the RT restarting fails to deliver a better accuracy as the tolerance gets smaller. This is despite increased computational costs (27 instead of 20 steps). The AccuRT implementation, with the same sparse LU factorization as the linear solver, is able to give the required accuracy, although the CPU time is increased by a factor of 10 (see line 3 of Table 1). Note that this CPU time measurement (done in MATLAB) is not very representative: it does not correspond to the number of steps (73 versus 27). This is because direct solvers (LU factorization and the backslash operator) are implemented in MATLAB quite efficiently, whereas the iterative solvers not. Hence, in the test problem the CPU time for applying the sparse LU factorization as a preconditioner within the GMRES(10) solver turns out to be relatively high. However, once the algorithm detects a suitable value of $\gamma$, this value can be used to compute other matrix exponential actions. In this case, as seen in line 4 of the Table 1, we get the same high accuracy for a moderate increase in the CPU time.

Furthermore, we evaluate our approach on this test problem with an iterative inner solver. For this purpose we used GMRES(10) with the ILUT($\varepsilon = 10^{-3}$) preconditioner. Line 5 of the Table 1 shows that the RT restarting combined with the preconditioned iterative solver requires 36 SAI-Arnoldi steps (instead of the 27 steps with the direct solver, see line 2 of the table). This is because the residuals in these two implementations of the method (with the iterative solver and with the direct solver) slightly differ. AccuRT with the same inner iterative setting requires roughly twice as much CPU time than RT restarting but the accuracy is now improved, see line 6. Finally, once the AccuRT has detected a proper value of $\gamma$, it allows to achieve a better accuracy with comparable computational costs.

In the lower part of Table 1 the results for the larger Peclet number $\text{Pe} = 1000$ are presented. For the tolerance $\text{tol} = 1e-06$ the RT restarting gives a result with an accuracy $3.36e-07$. This seems to be fine. However, for the tolerance $\text{tol} = 1e-08$ the attained accuracy is the same, while the computational costs increase from 14 to 27 steps. In the following two lines of the table we see the results for the AccuRT restarting. It does yield an improved accuracy for an increased CPU time. As seen in the next table line, once a proper value of $\gamma$ is detected, the same accuracy can be obtained within comparable CPU time.

### Table 1. Results for the convection-diffusion test problem, 802 × 802 grid, final time $t = 1$

| Method | Tolerance, error | CPU time, s | Steps (inner iterations GMRES(10)) |
|--------|------------------|-------------|-----------------------------------|
| RT, sparse LU | $1e-06$, $2.50e-07$ | 46.2 | 20 (–) |
| RT, sparse LU | $1e-08$, $2.51e-07$ | 48.0 | 27 (–) |
| AccuRT, sparse LU, GMRES(10) | $1e-08$, $1.60e-08$ | 484 | 73 (962) |
| AccuRT, sparse LU, GMRES(10), detected $\gamma$ | $1e-08$, $1.65e-08$ | 56.2 | 53 (–) |
| RT, GMRES(10)/ILUT | $1e-08$, $2.54e-07$ | 90.5 | 36 (440) |
| AccuRT, GMRES(10)/ILUT | $1e-08$, $1.85e-08$ | 191 | 77 (1258) |
| GMRES(10)/ILUT, detected $\gamma$ | $1e-08$, $1.51e-08$ | 68.2 | 57 (342) |
| Pe = 200, restart length 10 |         |             |                                  |
| RT, GMRES(10)/ILUT | $1e-06$, $3.36e-07$ | 49.2 | 14 (154) |
| RT, GMRES(10)/ILUT | $1e-08$, $3.52e-07$ | 72.9 | 27 (325) |
| AccuRT, GMRES(10)/ILUT | $1e-06$, $2.43e-07$ | 44.5 | 17 (136) |
| AccuRT, GMRES(10)/ILUT | $1e-08$, $7.55e-08$ | 110 | 55 (630) |
| AccuRT, GMRES(10)/ILUT, detected $\gamma$ | $1e-08$, $7.41e-08$ | 52.8 | 25 (205) |
| Pe = 1000, restart length 10 |         |             |                                  |
3.3. Maxwell’s Equations in a Lossless Medium

Consider the Maxwell equations for a three-dimensional domain in a lossless and source-free medium:

\[
\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu} \nabla \times \mathbf{E},
\]

\[
\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\varepsilon} \nabla \times \mathbf{H}.
\]  \tag{24}

Here, \(\varepsilon\) and \(\mu\) are scalar functions of \((x, y, z)\) called permittivity and permeability, respectively, whereas magnetic field \(\mathbf{H}\) and electric field \(\mathbf{E}\) are unknown vector-valued functions of \((x, y, z, t)\). The boundary conditions assign zero to the tangential electric field components. Physically this can mean either perfectly conducting domain boundary or the so-called “far field condition” \([41, 42]\). The problem setup is adopted from the second test in \([42]\): in a spatial domain \([-6.05, 6.05] \times [-6.05, 6.05] \times [-6.05, 6.05]\) filled with air (relative permittivity \(\varepsilon = 1\)) a dielectric specimen with relative permittivity \(\varepsilon_r = 5.0\) is placed which occupies the region \([-4.55, 4.55] \times [-4.55, 4.55] \times [-4.55, 4.55]\). In the specimen there are 27 spherical voids \((\varepsilon_r = 1)\) of radius 1.4, whose centers are positioned at \((x_i, y_j, z_k) = (3.03i, 3.03j, 3.03k), i, j, k = -1, 0, 1\). The initial values for all the components of both fields \(\mathbf{H}\) and \(\mathbf{E}\) are set to zero, except for the \(x-\) and \(y-\)components of \(\mathbf{E}\). These two have nonzero values in the middle of the spatial domain to represent a light emission.

The standard finite-difference staggered Yee discretization in space leads to an ODE system of the form \((3)\). The spatial meshes used in the test comprise either \(40 \times 40 \times 40\) or \(80 \times 80 \times 80\) grid Yee cells and lead to problem size \(\mathbf{v} \in \mathbb{R}^{n} = 413526\) or \(\mathbf{v} = 3188646\), respectively. After the discretization is carried out, the initial value vector \(\mathbf{v} \in \mathbb{R}^n\) is normalized as \(\mathbf{v} := \mathbf{v} / \|\mathbf{v}\|\). Comparison of the results obtained for the two meshes shows that this spatial resolution should be sufficient for this test. The final time is set to \(t = 1\).

This test represents a tough problem for the SAI Krylov method because the matrix \(\mathbf{A}\) is strongly nonsymmetric (in fact, a diagonal matrix \(\mathbf{D}\) can be chosen such that \(\mathbf{D}^{-1} \mathbf{A}\) is skew-symmetric). For strongly nonsymmetric problems, such as the discretized lossless Maxwell equations, SAI Krylov methods are likely to be inefficient \([43]\) and, indeed, other restarted Krylov subspace exponential schemes are reported to be more efficient for this test problem \([22]\). Furthermore, it is a three-dimensional vector problem where 6 unknowns \((x, y, z)\) vector components for each field) are associated with each computational cell. Hence, depending on the specific parameter values, solving linear systems with the shifted matrix \(\mathbf{I} + \gamma \mathbf{A}\) may not be a trivial task. However, for this particular test setting it turns out that \(\gamma \|\mathbf{A}\|\) is small enough so that condition \((23)\) does not hold and even unpreconditioned Richardson iteration can successfully be used for solving the shifted linear systems. In the test runs, we use unpreconditioned GMRES(10) iterative solver. Therefore, we include this test to show capabilities of the proposed AccuRT restarting.

Our experience indicates that, to have a fast convergence with SAI Krylov methods for strongly nonsymmetric matrices \(\mathbf{A}\), \(\gamma\) should be taken significantly smaller than the commonly used value \(t/10\) \([44]\). Hence we set \(\gamma\) to \(t/80 = 1/80\) for both RT and AccuRT methods, and the AccuRT restarting can eventually further decrease this value. The results are presented in Table 2 and in Figs. 5 and 6. The AccuRT restarting clearly outperforms the RT restarting both in terms the costs and the attained accuracy. The accuracy loss in RT restarting occurs at the first restarts when the residual norm turns out to be nowhere within the required tolerance. Avoid the accuracy loss, the AccuRT reduces \(\gamma\) which not only repairs the
Botchev

accuracy but also leads to a faster solution of the shifted systems (recall that the smaller $\gamma$, the faster unpre-
conditioned GMRES converges). Moreover, the reduced $\gamma$ leads then to a further efficiency gain by
AccuRT.

As can be seen in Fig. 6, it is obtained due to larger intervals $[0, \delta]$ on which the residual norm appears
to be smaller than the required tolerance (recall that the time interval at each restart reduces from $[0, t]$ to
$[0, t - \delta]$).

4. CONCLUSIONS

The presented AccuRT (accurate residual-time) restarting seems to be a promising approach for SAI
(shift-and-invert) Krylov subspace methods evaluating the matrix exponential of nonsymmetric matrices.

Fig. 5. Convergence of the SAI Krylov method with RT (solid line) and AccuRT (dashed line) restarting for lossless Max-
well equations test, mesh $40 \times 40 \times 40$, restart length 7. The bottom plot is a close up of the top plot. Each zigzag corre-
sponds to a restart.

Fig. 6. Restarting efficiency as a ratio of the decreased time interval length $\delta$ and total remaining time interval length $t$ for
the lossless Maxwell equations test, mesh $40 \times 40 \times 40$, restart length 7. Efficiency 0% at the second AccuRT restart
means that $\gamma$ is decreased.
It has all the attractive properties of the standard RT (residual-time) restarting and allows to avoid its accuracy loss while preserving efficiency of the method.

Several research directions for further studies can be indicated. First, the minimum search of the residual norm is currently carried out on a uniform set of points spread over the time interval. This should probably be better done on a nonuniform grid refined in the regions where the residual norm has its local minima. An adaptive procedure for building such a mesh could be designed. Furthermore, a question on how to extend this approach to symmetric matrices could be investigated. We hope to address these issues in the future.

FUNDING

This work is supported by the Russian Science Foundation grant no. 19-11-00338.

REFERENCES

1. M. Hochbruck and A. Ostermann, “Exponential integrators,” Acta Numer. 19, 209–286 (2010).
2. O. de la Cruz Cabrera, M. Matar, and L. Reichel, “Analysis of directed networks via the matrix exponential,” J. Comput. Appl. Math. 355, 182–192 (2019). https://doi.org/10.1016/j.cam.2019.01.015
3. P. Kürschner, “Balanced truncation model order reduction in limited time intervals for large systems,” Adv. Comput. Math. 44 (2018). https://doi.org/10.1007/s10444-018-9608-6
4. A. Frommer and V. Simoncini, “Matrix functions,” Model Order Reduction: Theory, Research Aspects and Applications, Ed. by W. H. Schilders, H. A. van der Vorst, and J. Rommes (Springer, Berlin, 2008), pp. 275–304.
5. L. Bergamaschi and M. Vianello, “Efficient computation of the exponential operator for large, sparse, symmetric matrices,” Numer. Linear Algebra Appl. 7 (1), 27–45 (2000).
6. A. H. Al-Mohy and N. J. Higham, “Computing the action of the matrix exponential, with an application to exponential integrators,” SIAM J. Sci. Comput. 33 (2), 488–511 (2011). https://doi.org/10.1137/100788860
7. I. Moret and P. Novati, “RD rational approximations of the matrix exponential,” BIT 44, 595–615 (2004).
8. J. van den Eshof and M. Hochbruck, “Preconditioning Lanczos approximations to the matrix exponential,” SIAM J. Sci. Comput. 27 (4), 1438–1457 (2006).
9. V. Druskin, L. Knizhnerman, and M. Zaslavsky, “Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts,” SIAM J. Sci. Comput. 31 (5), 3760–3780 (2009).
10. S. Güttel, PhD Thesis (Tech. Univ. Bergakademie, Freiberg, 2010). www.guettel.com.
11. S. Güttel, “Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection,” GAMM Mitteilungen 36 (1), 8–31 (2013). www.guettel.com.
12. E. Celledoni and I. Moret, “A Krylov projection method for systems of ODEs,” Appl. Numer. Math. 24 (2–3), 365–378 (1997).
13. H. Tal-Ezer, “On restart and error estimation for Krylov approximation of \( w = f(A)v \),” SIAM J. Sci. Comput. 29 (6), 2426–2441 (2007). https://doi.org/10.1137/040617868
14. M. Afanasjew, M. Eiermann, O. G. Ernst, and S. Güttel, “Implementation of a restarted Krylov subspace method for the evaluation of matrix functions,” Linear Algebra Appl. 429, 2293–2314 (2008).
15. M. Eiermann, O. G. Ernst, and S. Güttel, “Deflated restarting for matrix functions,” SIAM J. Matrix Anal. Appl. 32 (2), 621–641 (2011).
16. A. Frommer, S. Güttel, and M. Schweitzer, “Efficient and stable Arnoldi restarts for matrix functions based on quadrature,” SIAM J. Matrix Anal. Appl. 35 (2), 661–683 (2014).
17. T. Jawecki, W. Auzinger, and O. Koch, “Computable strict upper bounds for Krylov approximations to a class of matrix exponentials and \( \phi \)-functions,” (2018). https://arxiv.org/pdf/1809.03369.
18. M. Hochbruck and C. Lubich, “Exponential Integrators for quantum-classical molecular dynamics,” BIT 39 (4), 620–645 (1999).
19. M. Hochbruck, T. Pažur, A. Schulz, et al., “Efficient time integration for discontinuous Galerkin approximations of linear wave equations,” Z. Angew. Math. Mech. 95 (3), 237–259 (2015). https://doi.org/10.1002/zamm.201300306
20. R.-U. Börner, O. G. Ernst, and S. Güttel, “Three-dimensional transient electromagnetic modeling using rational Krylov methods,” Geophys. J. Int. 202 (3), 2025–2043 (2015).
21. M. A. Botchev, A. M. Hanse, and R. Uppu, “Exponential Krylov time integration for modeling multifrequency optical response with monochromatic sources,” J. Comput. Appl. Math. 340, 474–485 (2018). https://doi.org/10.1016/j.cam.2017.12.014
22. M. A. Botchev and L. A. Knizhnerman, “ART: Adaptive residual-time restarting for Krylov subspace matrix exponential evaluations,” J. Comput. Appl. Math. **364** (112311) (2020). https://doi.org/10.1016/j.cam.2019.06.027

23. J. Niehoff, PhD Thesis (Mathematisch-Naturwissenschaftlichen Fakultät der Heinrich-Heine-Universität Düsseldorf, 2006).

24. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. (Johns Hopkins Univ. Press, Baltimore, 1996).

25. C. B. Moler and C. F. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,” SIAM Rev. **45** (1), 3–49 (2003).

26. N. J. Higham, *Functions of Matrices: Theory and Computation* (SIAM, Philadelphia, PA, 2008).

27. M. A. Botchev, V. Grimm, and M. Hochbruck, “Residual, restarting, and Richardson iteration for the matrix exponential,” SIAM J. Sci. Comput. A **35** (3), 1376–1397 (2013). https://doi.org/10.1137/110820191

28. B. N. Parlett, *The Symmetric Eigenvalue Problem* (SIAM, Philadelphia, PA, 1998).

29. H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems* (Cambridge Univ. Press, Cambridge, 2003).

30. Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2d ed. (SIAM, Philadelphia, PA, 2003). http://www-users.cs.umn.edu/~saad/books.html.

31. V. L. Druskin and L. A. Knizhnerman, “Two polynomial methods of calculating functions of symmetric matrices,” USSR Comput. Math. Math. Phys. **29** (6), 112–121 (1989).

32. L. A. Knizhnerman, “Calculation of functions of unsymmetric matrices using Arnoldi’s Method,” USSR Comput. Math. Math. Phys. **31** (1), 1–9 (1991).

33. M. Hochbruck and C. Lubich, “On Krylov subspace approximations to the matrix exponential operator,” SIAM J. Numer. Anal. **34** (5), 1911–1925 (1997).

34. V. L. Druskin, A. Greenbaum, and L. A. Knizhnerman, “Using nonorthogonal Lanczos vectors in the computation of matrix functions,” SIAM J. Sci. Comput. **19** (1), 38–54 (1998).

35. W. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations* (Springer, Berlin, 2003).

36. R. Barrett, M. Berry, T. F. Chan, et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (SIAM, Philadelphia, PA, 1994). www.netlib.org/templates.

37. Y. Saad and M. H. Schultz, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” SIAM J. Sci. Stat. Comput. **7** (3), 856–869 (1986).

38. Y. Saad, *Iterative Methods for Sparse Linear Systems*, Book out of print (2000). www-users.cs.umn.edu/~saad/books.html.

39. R. B. Sidje, “Expokit: A software package for computing matrix exponentials,” ACM Trans. Math. Software **24** (1), 130–156 (1998). www.maths.uq.edu.au/expokit.

40. L. A. Krukier, “Implicit difference schemes and an iterative method for solving them for a certain class of systems of quasi-linear equations,” Sov. Math. **23** (7), 43–55 (1979).

41. A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3rd ed. (Artech House, Boston, MA, 2005).

42. J. S. Kole, M. T. Figge, and H. De Raedt, “Unconditionally stable algorithms to solve the time-dependent Maxwell equations,” Phys. Rev. E **64**, 066705 (2001).

43. J. G. Verwer and M. A. Botchev, “Unconditionally stable integration of Maxwell’s equations,” Linear Algebra Appl. **431** (3–4), 300–317 (2009).

44. M. A. Botchev, “Krylov subspace exponential time domain solution of Maxwell’s equations in photonic crystal modeling,” J. Comput. Appl. Math. **293**, 24–30 (2016). https://doi.org/10.1016/j.cam.2015.04.022