



ART: Adaptive residual-time restarting for Krylov subspace matrix exponential evaluations



M.A. Botchev^{a,*}, L.A. Knizhnerman^b

^a Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Miusskaya Sq. 4, 125047 Moscow, Russia

^b Mathematical Modelling Department, Central Geophysical Expedition, Narodnogo Opolcheniya St., 38, bldg. 3, 123298 Moscow, Russia

ARTICLE INFO

Article history:

Received 25 December 2018

Received in revised form 13 June 2019

MSC:

65F60

65F30

65N22

65L05

35Q61

Keywords:

Krylov subspace methods

Exponential time integration

Arnoldi method

Krylov subspace restarting

Shift-and-invert Krylov subspace methods

ABSTRACT

In this paper a new restarting method for Krylov subspace matrix exponential evaluations is proposed. Since our restarting technique essentially employs the residual, some convergence results for the residual are given. We also discuss how the restart length can be adjusted after each restart cycle, which leads to an adaptive restarting procedure. Numerical tests are presented to compare our restarting with three other restarting methods. Some of the algorithms described in this paper are a part of the Octave/Matlab package expmARPACK available at <http://team.kiam.ru/botchev/expm/>.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Computing actions of the matrix exponential for large usually sparse matrices is an important task occurring e.g. in time integration of ODE (ordinary differential equation) systems, network analysis, Markov chain models and many other problems. Krylov subspace methods appear to be an indispensable tool to carry out this task, especially for nonsymmetric or stiff matrices. Other methods applicable to this task include polynomial interpolation methods [1–3], stabilized explicit time stepping methods [4–6] and scaled truncated Taylor series approximations [7].

To be computationally efficient, Krylov subspace methods for evaluating actions of the matrix exponential on given vectors often have to rely on so-called restarting techniques [8]. Restarting techniques allow to decrease memory and computational expenses of the Krylov subspace methods by storing (and working with) only a limited, restricted number of the Krylov subspace basis vectors, in contrast to the nonrestarted methods whose memory and work costs usually grow with the iteration number. We emphasize that for general matrix function evaluations no short-term recurrence Krylov subspace methods exist (such as the conjugate gradient method for solving linear systems). On the other hand, the design of Krylov subspace restarting techniques for general matrix functions is a more complicated task than for solving linear systems [9].

* Corresponding author.

E-mail addresses: botchev@ya.ru (M.A. Botchev), lnizhnerman@gmail.com (L.A. Knizhnerman).

Different restarting techniques for the Krylov subspace methods to evaluate matrix exponential actions have been developed. In the EXPOKIT package [10], the restarting is based on the division of the time interval $[0, t]$ into smaller intervals to facilitate Krylov subspace convergence. Here, $t > 0$ is the time at which the solution $y(t) = \exp(-tA)v$ to the associated initial value problem (IVP) (2) has to be computed. Authors of [11] propose restarting which employs a residual concept (cf. (7)): the solution of the next restart is an approximate correction with respect to the residual. This residual restarting is further developed and tested in [12]. In [13, Chapter 3] the proposed restarting is based on the so-called generalized residual [14] and on the analogy between Krylov subspace methods for solving linear systems and for evaluating matrix exponential actions. Another restarting is proposed in [9]: it uses the observation that the Krylov subspace methods for evaluating $f(A)v$, with $f(\cdot)$ being a given matrix function and v a given vector, can be seen as polynomial methods which interpolate f at the so-called Ritz values [15]. Other restarting techniques include [16–20].

In this paper we present a new restarting procedure for Krylov subspace methods which is applicable to the matrix exponential. Our restarting technique has an attractive property, namely, convergence of the restarted Krylov subspace method to the sought after solution $e^{-tA}v$ is guaranteed (though can be slow) for any length of the restart. Since our restarting procedure relies on the behavior of the residual (cf. (7)), we provide some convergence estimates for the residual norm. We also show how the restarting we present here can be extended to an adaptive procedure to choose a good length for the next restart. Numerical tests are presented showing the potential of our adaptive restarting. We also discuss how this restarting can be combined with the shift-and-invert (SAI) rational Krylov subspace approximations. Another contribution of our paper is that the new restarting is tested numerically together with three other restarting methods: the time step restarting of EXPOKIT [10], the Niehoff–Hochbruck restarting [13] and the residual restarting [11,12].

The paper is organized as follows. In Section 2 some preliminaries on the Krylov subspace approximations to the matrix exponential and their residuals are given, as well as some results on residual convergence. Furthermore, in this section the new restarting procedure is discussed and an adaptive way to choose an optimal Krylov dimension based on this restarting procedure is introduced. In this section we also discuss how the restarting can be generalized to the shift-and-invert (SAI) Krylov subspace approximations. In Section 3 we describe numerical experiments and present their results. Finally, the conclusions are drawn in Section 4.

2. Restarting procedure

2.1. Preliminaries

Throughout the paper, unless indicated otherwise, $\|\cdot\|$ denotes the 2-norm. We also assume that $A \in \mathbb{R}^{n \times n}$ is a matrix such that

$$\operatorname{Re}(x^*Ax) \geq 0, \quad \forall x \in \mathbb{C}^n, \quad (1)$$

where $\operatorname{Re}(z)$ denotes the real part of a complex number z . This means that initial value problem

$$y'(t) = -Ay(t), \quad y(0) = v, \quad (2)$$

with $v \in \mathbb{R}^n$ given, is well posed, see e.g. [21]. Moreover, (1) implies that

$$\|\exp(-tA)\| \leq e^{-t\omega}, \quad t \geq 0, \quad (3)$$

where $-\omega = \mu(-A) \leq 0$ is the logarithmic 2-norm of the matrix $-A$ [21,22]. Let $\mathcal{K}_k(A, v)$ be the Krylov subspace

$$\mathcal{K}_k(A, v) = \operatorname{span}\{v, Av, \dots, A^{k-1}v\}.$$

In practice, the Arnoldi process (or, if A is (possibly skew) Hermitian, the Lanczos process) is usually exploited to compute an orthonormal basis v_1, \dots, v_k of $\mathcal{K}_k(A, v)$ (see for instance [23–26]):

$$V_k = [v_1 \ \dots \ v_k] \in \mathbb{R}^{n \times k}, \quad \operatorname{colspan}(V_k) = \mathcal{K}_k(A, v), \quad (4)$$

$$AV_k = V_{k+1}H_k,$$

where $H_k \in \mathbb{R}^{(k+1) \times k}$ is an upper Hessenberg matrix. The relation $AV_k = V_{k+1}H_k$ is called the Arnoldi decomposition. Using the fact that the last row of H_k contains a single nonzero entry $h_{k+1,k} \geq 0$, one can rewrite (4) as

$$AV_k = V_{k+1}H_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T, \quad (5)$$

where $H_k = V_k^T AV_k$ is the matrix H_k with the last row skipped and $e_k = (0, \dots, 0, 1)^T \in \mathbb{R}^k$. A Krylov subspace approximation $y_k(t)$ to $y(t) = \exp(-tA)v$ can be computed as [15,27–30]

$$y_k(t) = V_k \exp(-tH_k)(\beta e_1), \quad (6)$$

where $e_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^k$ and $\beta = \|v\|$.

A convenient way to monitor the error of the Krylov subspace approximation $y_k(t)$ is to check its residual with respect to the ODE in (2)

$$r_k(t) = -y'_k(t) - Ay_k(t), \quad (7)$$

which is readily computable as [11,12,31]

$$r_k(t) = \beta_k(t)v_{k+1}, \quad \beta_k(t) = -h_{k+1,k}e_k^T \exp(-tH_k)(\beta e_1). \quad (8)$$

Although this residual notion for the matrix exponential is known and used in the literature (for example, to obtain upper bounds on the error, see [31, formula (32)] and [12, Lemma 4.1]), there are hardly any convergence estimates for $r_k(t)$ available. Therefore, since we essentially use this exponential residual in this paper, we now first give a general a priori convergence result for its norm.

2.2. An estimate on the residual in terms of the Faber series

Faber series as a means to investigate convergence of Arnoldi method are introduced in [29]; see also [32]. Let $W(A) = \{x^*Ax \mid x \in \mathbb{C}^n, \|x\| = 1\}$ be the numerical range of A and let Φ_j be the Faber polynomials [33] on the compact $W(A)$. Consider the Faber series decomposition

$$f(z; t) = e^{-tz} = \sum_{j=0}^{+\infty} f_j(t)\Phi_j(z), \quad z \in W(A), \quad t \geq 0, \quad (9)$$

where t is considered as a parameter.

Proposition 1. *The residual of the Arnoldi method $r_k(t)$ defined by (7) satisfies the inequality*

$$\|r_k(t)\| \leq 2h_{k+1,k} \sum_{j=k-1}^{+\infty} |f_j(t)| \leq 2\|A\| \sum_{j=k-1}^{+\infty} |f_j(t)|. \quad (10)$$

Proof. Throughout the proof for simplicity we omit the index \cdot_k in H_k and V_k . The Faber series converges super-exponentially in j and its coefficients $f_j(t)$ are smooth in t [33, Chapter 3]. This enables us to differentiate series (9) in t .

Decomposition (9) induces the decomposition of the approximant

$$y_k = Vf(H; t)e_1 = V \sum_{j=0}^{+\infty} f_j(t)\Phi_j(H)e_1. \quad (11)$$

Evidently,

$$zf(z; t) + \frac{\partial f(z; t)}{\partial t} = 0,$$

whence

$$0 = z \sum_{j=0}^{+\infty} f_j(t)\Phi_j(z) + \sum_{j=0}^{+\infty} f_j'(t)\Phi_j(z) = \sum_{j=0}^{+\infty} [f_j(t)z + f_j'(t)]\Phi_j(z). \quad (12)$$

Exploiting (7), (11), (4), (12) with H substituted for z , and the equality $\deg \Phi_j = j$, derive

$$\begin{aligned} -r_k(t) &= AV \sum_{j=0}^{+\infty} f_j(t)\Phi_j(H)e_1 + V \sum_{j=0}^{+\infty} f_j'(t)\Phi_j(H)e_1 \\ &= (VH + h_{k+1,k}q_{k+1}e_k^T) \sum_{j=0}^{+\infty} f_j(t)\Phi_j(H)e_1 + V \sum_{j=0}^{+\infty} f_j'(t)\Phi_j(H)e_1 \\ &= V \sum_{j=0}^{+\infty} [f_j(t)H + f_j'(t)]\Phi_j(H)e_1 + h_{k+1,k}q_{k+1}e_k^T \sum_{j=0}^{+\infty} f_j(t)\Phi_j(H)e_1 \\ &= h_{k+1,k}q_{k+1}e_k^T \sum_{j=0}^{+\infty} f_j(t)\Phi_j(H)e_1 = h_{k+1,k}q_{k+1}e_k^T \sum_{j=k-1}^{+\infty} f_j(t)\Phi_j(H)e_1. \end{aligned}$$

The bound $\|\Phi_j(H)\| \leq 2$ (see [34, theorem 1]) now implies (10). \square

Remark 1. For rendering $|f_j(t)|$ specific see [32, section 4].

Remark 2. Comparing of our estimate (10) with the estimate [32, theorem 3.2] for the error

$$\|y(t) - y_k(t)\| \leq 4 \sum_{j=k}^{+\infty} |f_j(t)|,$$

we see that these two upper bounds differ mainly in coefficients. This gives one a hope that the error and the residual behave similarly to each other. We also note that there exist error bounds in terms of the residual (see [31, formula (32)] and [12, Lemma 4.1]).

We now discuss our restarting procedure.

2.3. Restarting algorithm

The restarting procedure we propose here is based on the observation that, even for very small values of k , the residual $r_k(s)$ is small in norm on some interval $s \in [0, \delta]$ where δ is taken sufficiently small. This is formulated more precisely in the following statement (which is a simple result given here for completeness). Let us define function $\varphi_1(z)$ as (see e.g. [35])

$$\varphi_1(z) = (e^z - 1)/z.$$

Lemma 1. Let y_k be the Krylov approximation (6) to $y(t) = \exp(-tA)v$ and let k be fixed. Then for the residual $r_k(t)$ defined by (8) holds

$$\|r_k(t)\| \leq th_{k+1,k} \|H_k\| \beta \varphi_1(-t\omega_k),$$

where $-\omega_k = \mu(-H_k)$ is the logarithmic norm of $-H_k$. Hence, for any k and $\varepsilon > 0$ there exists a $\delta_k > 0$ such that

$$\|r_k(s)\| \leq \varepsilon, \quad \forall s \in [0, \delta_k].$$

Proof. It follows from (8) that

$$\|r_k(t)\| = |\beta_k(t)| = h_{k+1,k} \cdot |e_k^T u(t)|,$$

where $u(t) = \exp(-tH_k)(\beta e_1)$ solves the initial value problem

$$u'(t) = -H_k u(t), \quad u(0) = \beta e_1. \quad (13)$$

Then we have

$$u(t) - u(0) = (\exp(-tH_k) - I)u(0) = -tH_k \varphi_1(-tH_k)u(0).$$

Since $H_k = V_k^T A V_k$ is a Galerkin projection of A , from (1) it follows that

$$\begin{aligned} \lambda_{\min} \left(\frac{1}{2}(A + A^T) \right) &\leq \lambda_{\min} \left(\frac{1}{2}(H_k + H_k^T) \right), \\ \lambda_{\max} \left(\frac{1}{2}(H_k + H_k^T) \right) &\leq \lambda_{\max} \left(\frac{1}{2}(A + A^T) \right), \end{aligned} \quad (14)$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimal and maximal eigenvalues, respectively. We further note that

$$\lambda_{\min} \left(\frac{1}{2}(A + A^T) \right) = \omega,$$

where ω is the constant defined in (3) (see e.g. [21,22]) and denote $\omega_k = \lambda_{\min}(\frac{1}{2}(H_k + H_k^T))$. Hence, H_k inherits the property (3) of A , so that

$$\|\exp(-tH_k)\| \leq e^{-t\omega_k}, \quad t \geq 0, \quad (15)$$

with $-\omega_k = \mu(-H_k) \leq -\omega$. Using the estimate in the proof of Lemma 2.4 in [35], one can see that (3) implies

$$\|\varphi_1(-tA)\| \leq \varphi_1(-t\omega), \quad (16)$$

so that a similar relation holds for $\|\varphi_1(-tH_k)\|$. Hence,

$$\begin{aligned} |e_k^T u(t)| &= |e_k^T (u(t) - u(0))| \leq \|u(t) - u(0)\| \\ &\leq t \|H_k\| \|\varphi_1(-tH_k)\| \|u(0)\| \leq t \|H_k\| \beta \varphi_1(-t\omega_k). \end{aligned} \quad (17)$$

Then the last part of the Lemma statement follows from the observation that $0 < \varphi_1(z) \leq 1$ for $z \leq 0$. \square

```

% Given:   $t > 0$ ,  $\text{tol} > 0$ ,  $\underline{H}_k$  (cf. (4)-(6))
% determine  $n_t$  - number of time steps to monitor the residual
 $u := e_1$ 
 $n_t := 100$  % initial value of  $n_t$ 
 $r := 2\text{tol}$ 
while  $r > \text{tol}$ 
     $E := \exp(-(t/n_t)H_k)$ 
     $u := Ee_1$ 
     $r := -h_{k+1,k}(e_k^T u)$ 
    if  $|r| \leq \text{tol}$ 
         $u := e_1$ 
        break the while loop
    end
     $n_t := 2n_t$ 
end
% compute residual for intermediate time points
% until it exceeds tolerance
for  $i = 1, \dots, n_t$ 
     $u_0 := u$ 
     $u := Eu$ 
     $r := -h_{k+1,k}(e_k^T u)$ 
    if  $|r| > \text{tol}$ 
         $\delta_k := \frac{i-1}{n_t}t$ 
         $u := u_0$ 
        break the for loop
    end
end
end

```

Fig. 1. This algorithm determines δ_k such that $\|r_k(s)\| \leq \beta \text{tol}$ for $s \in [0, \delta_k]$ and $u \in \mathbb{R}^k$ such that $y_k(\delta_k) = V_k u$. The initial value of n_t can be changed if desired.

It is not difficult to see that we cannot expect the estimate of Lemma 1 to be sharp. Indeed, the sharpness is lost in the estimates (17) and this is confirmed by numerical experiments given below in this section (see also Fig. 6). We now describe how our restarting procedure works and give some numerical illustrations.

Let tol be the given residual tolerance, i.e., we need to compute a Krylov subspace approximation $\tilde{y}(t)$ to $\exp(-tA)v$ whose residual \tilde{r} satisfies $\|\tilde{r}(s)\| \leq \text{tol}$ for all $s \in [0, t]$. Furthermore, let k_{\max} be the largest possible Krylov dimension, so that the costs for computing V_{k+1} , \underline{H}_k and storing V_{k+1} are unacceptably high for $k > k_{\max}$. Denote by δ_k the length of the interval $[0, \delta_k] \ni s$ such that $\|r_k(s)\| \leq \text{tol}$. We carry out (at most) k_{\max} steps of the Krylov method, computing on the way at every step k the residual norm $\|r_k(s)\|$ for several values $s \in [0, t]$ (including the value $s = t$, of course). If at step k the largest of the computed values $\|r_k(s)\|$ is below the given tolerance tol then we stop at this step (in this case no restarting is needed). Otherwise, if k_{\max} steps are done but the stopping criterion is not satisfied then we carry out our time restarting procedure. This means that we first determine the value $\delta_{k_{\max}}$. Taking into account Lemma 1, this is not a difficult task which is carried out by tracking the values of $\|r_k(s)\|$ for increasing s (this procedure is outlined in detail in Fig. 1). Then we set $v := y_{k_{\max}}(\delta_{k_{\max}})$ and solve the IVP (2) on a shorter time interval $[0, t - \delta_{k_{\max}}]$. Again, when solving this IVP on the shortened time interval, we can apply the same restarting procedure. We call this restarting method RT restarting (residual-time restarting) to emphasize its essential dependence on the time behavior of the residual function $r_k(s)$. The RT restarting algorithm is outlined in Fig. 2 and schematically illustrated in Fig. 3.

It is easy to see that our RT restarting procedure is guaranteed to converge for any restart length k_{\max} because there always exists a value $\delta_{k_{\max}} > 0$ such that the residual is sufficiently small in norm on the interval $[0, \delta_{k_{\max}}]$ (see Lemma 1).

To show how the restarting procedure works, in Fig. 4 we give convergence plots of the restarted Arnoldi method applied to the convection-diffusion test problem described in Section 3.1. The figure also shows the values of $\delta_{k_{\max}}$ plotted against the restart numbers. In addition, in Fig. 5 we plot the residual norms $\|r_k(s)\|$ versus $s \in [0, t]$. At this point it is instructive to demonstrate the estimate of Lemma 1. Therefore, in Fig. 6 the estimates are plotted which correspond to

```

% Given:  $A \in \mathbb{R}^{n \times n}$ ,  $v \in \mathbb{R}^n$ ,  $t > 0$ ,  $k_{\max}$  and  $\text{tol} > 0$ 
convergence := false
while not(convergence)
     $\beta := \|v\|$ ,  $v_1 := \frac{1}{\beta}v$ 
    for  $k = 1, \dots, k_{\max}$ 
         $w := Av_k$ 
        for  $i = 1, \dots, k$ 
             $h_{i,k} := w^T v_i$ 
             $w := w - h_{i,k}v_i$ 
        end
         $h_{k+1,k} := \|w\|$ 
         $s := (t/6)(1, 2, 3, 4, 5, 6)^T \in \mathbb{R}^6$ 
        for  $q = 1, \dots, 6$ 
             $u := \exp(-s_q H_k)e_1$ 
             $r(s_q) := -h_{k+1,k}(e_k^T u)$ 
        end
         $\text{rnorm} := \max\{r(s_1), \dots, r(s_6)\}$ 
        if  $\text{rnorm} \leq \text{tol}$ 
            convergence := true
            break
        elseif  $k = k_{\max}$ 
            % ----- restart after  $k_{\max}$  steps
            carry out Algorithm in Figure 1:
            find  $\delta_k$  and  $u = \exp(-\delta_k H_k)e_1$ 
             $v := V_k(\beta u)$ 
             $t := t - \delta_k$ 
            break
        end
         $v_{k+1} := \frac{1}{h_{k+1,k}}w$ 
    end
end
 $y_k := V_k(\beta u)$ 

```

Fig. 2. Description of the RT (residual-time) restarting algorithm. The algorithm computes Krylov subspace approximation $y_k(t) \approx \exp(-tA)v$ such that for its residual $r_k(t)$ holds $\|r_k(s)\| \leq \text{tol}$ for $s = t$ and several representative points $s \in (0, t)$.

the residuals shown in Fig. 5. We see that the upper bounds are, as expected, by no means sharp but they do reflect the time dependence of the residual norm $\|r_k(s)\|$, $s \in [0, t]$, qualitatively well.

2.4. Making the restarting procedure adaptive

As can be seen in numerical experiments (see right plots in Fig. 4), the value $\delta_{k_{\max}}$ tends to remain approximately constant after several first restarts. At each restart the time interval is decreased from $[0, t]$ to $[0, t - \delta_{k_{\max}}]$. Hence, at the end of each restart we can estimate the number of restarts that have yet to be done as $\approx t/\delta_{k_{\max}}$.

This observation allows to make our time restarting procedure adaptive as follows. Denote by $[\cdot]$ the rounding operation to the nearest integer. While carrying out Krylov steps $k = 1, \dots, k_{\max}$, we compute for several values k (in our experiments these are $[k_{\max}/3]$, $[2k_{\max}/3]$, $[5k_{\max}/6]$ and k_{\max}) the values δ_k and measured CPU times t_k^{cpu} spent to carry out these k steps. Then the values

$$\frac{t}{\delta_k} t_k^{\text{cpu}}, \quad k = [k_{\max}/3], [2k_{\max}/3], [5k_{\max}/6], k_{\max}$$

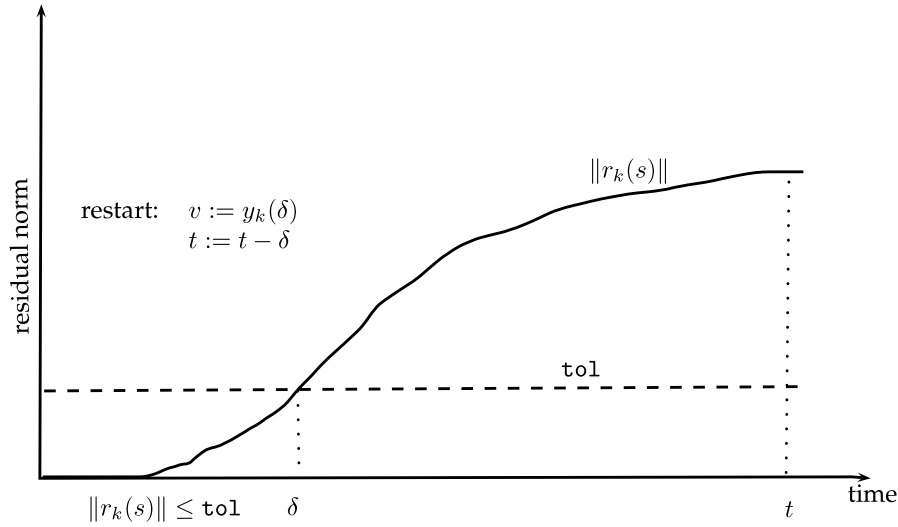


Fig. 3. A sketch of the RT restarting procedure.

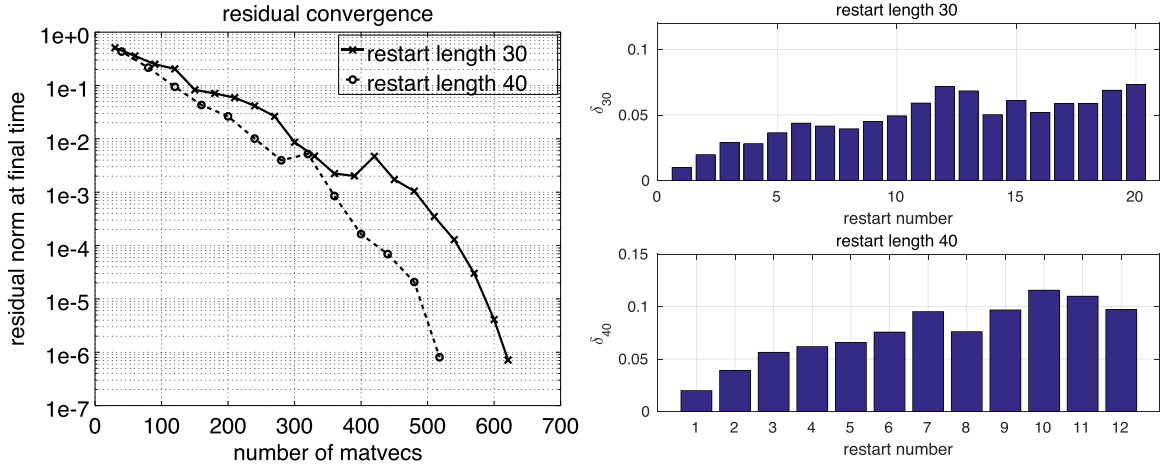


Fig. 4. Residual convergence plot (left) and the values of $\delta_{k_{\max}}$ against the restart numbers (right) for the convection-diffusion test problem (see Section 3.1). The matrix A is a discretized convection-diffusion operator for Peclet number $Pe = 100$, mesh 102×102 ($n = 100^2$).

are estimates of the remaining CPU needed to finish the computations for the restart lengths $[k_{\max}/3]$, $[2k_{\max}/3]$, $[5k_{\max}/6]$ and k_{\max} , respectively. Having computed these values during the restart, we can adjust the restart length for the next restart to have the smallest expected CPU time. In the experiments shown below we do so only if the expected gain in CPU time exceeds 5%. This adaptation procedure is then carried out at the end of every restart and also includes an option for an adaptive increase of the restart length as follows. If the current restart length is \tilde{k} , with $\tilde{k} < k_{\max}$, and the CPU time estimations indicate that the restart length \tilde{k} is the optimal one then the new restart length is set to $\min\{\tilde{k} + 5, k_{\max}\}$. We call this adaptive restarting ART: adaptive residual-time restarting. In Section 3 we test the ART restarting numerically and discuss it further.

2.5. RT restarting for the SAI Krylov subspace method

Shift-and-invert (SAI) Krylov subspace methods for computing actions of the matrix exponential are rational Krylov subspace methods designed to have a much faster convergence (in terms of the Krylov subspace dimension) than in regular polynomial Krylov subspace methods [36,37]. In SAI methods the Krylov subspace is built up for the matrix $(I + \gamma A)^{-1}$, $\gamma > 0$, rather than for A , and thus the price to pay for the faster convergence is solution, at each Krylov step, of linear systems with the matrix $I + \gamma A$. If these linear systems are solved by an iterative method, there are efficient strategies

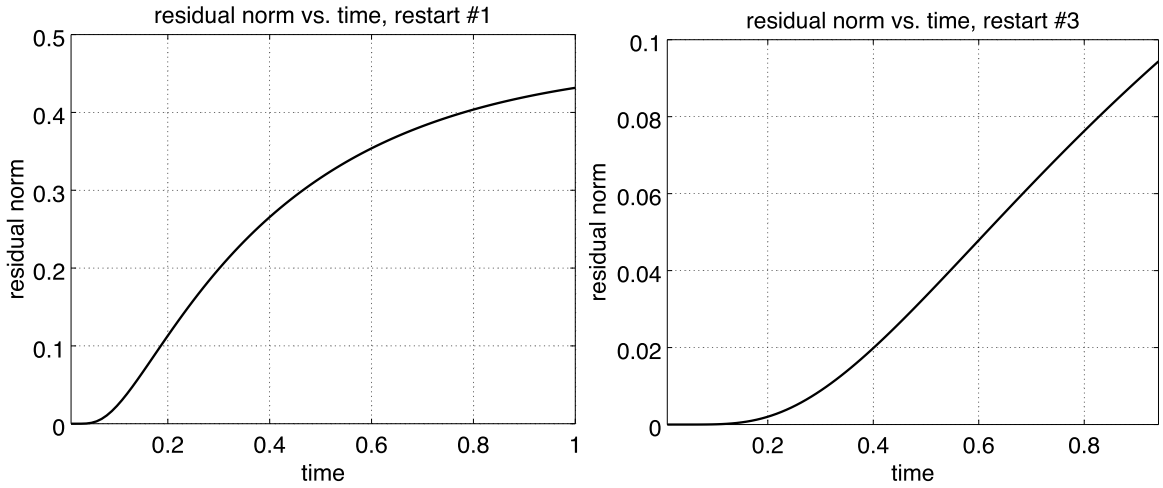


Fig. 5. Residual norms $\|r_k(s)\|$ versus $s \in [0, t]$ at the end of the first (left) and the third (right) restart cycles of length 40 for the convection–diffusion test problem (see Section 3.1). The time interval at right is shorter, as it has been decreased at first two restarts. The matrix A is a discretized convection–diffusion operator for Peclet number $Pe = 100$, mesh 102×102 ($n = 100^2$).

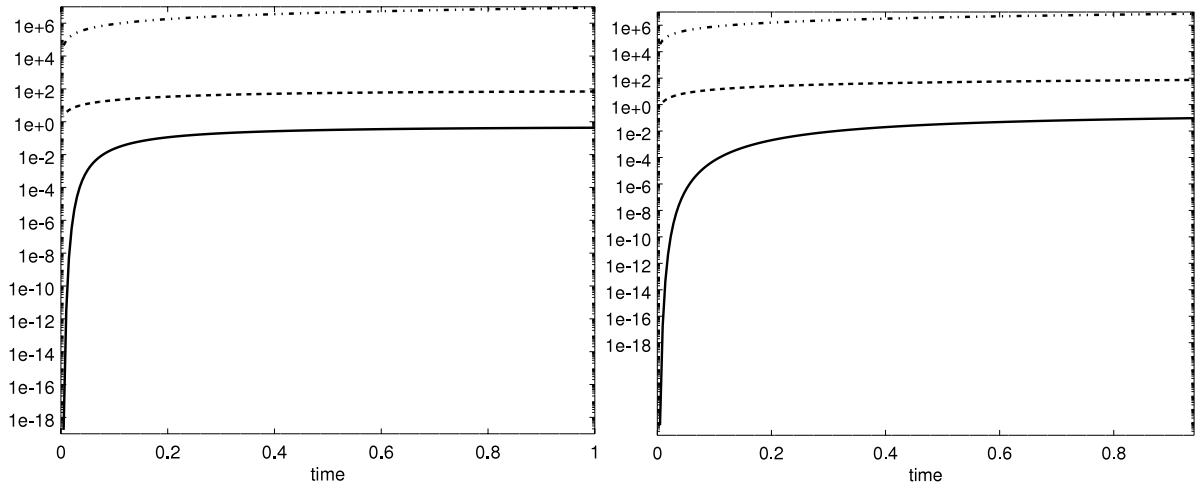


Fig. 6. Residual norms $\|r_k(s)\|$ versus $s \in [0, t]$ (left and right, solid lines) as shown in Fig. 5, together with their upper bounds given by Lemma 1 and estimates (17). Dashed lines: $h_{k+1,k}\|u(s) - u(0)\|$. Dashed-dotted and dotted lines (coinciding): $s\beta h_{k+1,k}\|H_k\| \|\varphi_1(-sH_k)\|$ and $s\beta h_{k+1,k}\|H_k\| \|\varphi_1(-s\omega_k)\|$.

to save computational work by relaxing the tolerance to which the systems are solved [37]. The shift γ can be chosen depending on the required tolerance tol , and for the tolerances of order 10^{-5} to 10^{-7} a good value of γ is $t/10$ where t is the time interval length [37]. This is the value we use in all our experiments. An attractive property of the SAI Krylov methods is their often observed space-mesh independent convergence, which can be proven for the discretizations of parabolic PDEs with a numerical range close to the positive real axis [37,38].

We now describe how the RT restarting strategy described above can be applied within the SAI Krylov subspace methods. The regular Arnoldi decomposition (4), (5) still holds for the SAI methods, namely

$$(I + \gamma A)^{-1} V_k = V_{k+1} \tilde{H}_k = V_k \tilde{H}_k + \tilde{h}_{k+1,k} v_{k+1} e_k^T, \quad (18)$$

where $\tilde{H}_k \in \mathbb{R}^{k \times k}$ denotes the first k rows of \tilde{H}_k . The approximation $y_k(t) \approx \exp(-tA)v$ is computed according to (6), with H_k being the SAI back transformed projection:

$$H_k = \frac{1}{\gamma} (\tilde{H}_k^{-1} - I). \quad (19)$$

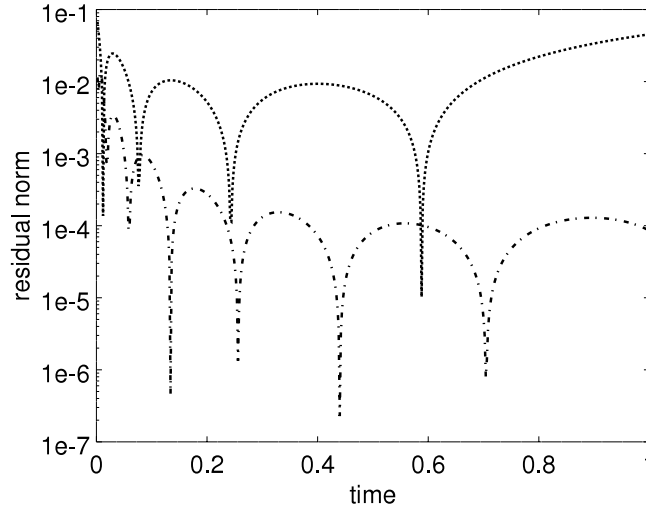


Fig. 7. Residual norms $\|r_k(s)\|$ versus $s \in [0, t]$ of the SAI Krylov subspace method after $k = 5$ (dotted line) and $k = 10$ (dash-dotted line) Krylov steps for the convection–diffusion test problem (see Section 3.1). The matrix A is a discretized convection–diffusion operator for Peclet number $Pe = 100$, mesh 102×102 ($n = 100^2$).

After rewriting the SAI Arnoldi decomposition (18) as [37, formula (4.1)]

$$AV_k = V_k H_k - \frac{\tilde{h}_{k+1,k}}{\gamma} (I + \gamma A) v_{k+1} e_k^T \tilde{H}_k^{-1}, \quad (20)$$

we can see that the residual $r_k(t) = -y'_k(t) - Ay_k(t)$ in the SAI Krylov subspace method reads [12]

$$r_k(t) = \beta_k(t)(I + \gamma A)v_{k+1}, \quad \beta_k(t) = \beta \frac{\tilde{h}_{k+1,k}}{\gamma} e_k^T \tilde{H}_k^{-1} \exp(-tH_k)e_1. \quad (21)$$

This residual function $r_k(s)$ has quite a different dependence on $s \in [0, t]$ than the residual in the regular Krylov method (8): its convergence $\|r_k(s)\| \rightarrow 0$ with growing k is much more uniform in $s \in [0, t]$. This means that we cannot expect that $r_k(s)$ is much smaller in norm for small s than for larger s . Typical dependence of the SAI residual norm $\|r_k(s)\|$ on $s \in [0, t]$ is shown in Fig. 7. As we see, there are a few distinct points s on the interval $[0, t]$ where $\|r_k(s)\|$ has values much smaller than the average value $\frac{1}{t} \int_0^t \|r_k(s)\| ds$. Therefore, we can carry out the RT restarting procedure by assigning δ_k the value s at which $\|r_k(s)\|$ has the smallest value:

$$\delta_k := \arg \min_{s \in [0, t]} \|r_k(s)\|.$$

The rest of the RT restarting is carried out exactly in the same way as for regular Krylov subspace methods, i.e., we set $v := y_{k_{\max}}(\delta_{k_{\max}})$ and proceed further by solving the IVP (2) on a shorter time interval $[0, t - \delta_{k_{\max}}]$. The drawback of this approach is that the reachable accuracy is restricted by the value $\|r_k(\delta_k)\| = \min_{s \in [0, t]} \|r_k(s)\|$ (this is the error introduced to the initial data of the restarted IVP by setting $v := y_{k_{\max}}(\delta_{k_{\max}})$). For instance, as can be seen in Fig. 7, for the restart length $k_{\max} = 5$ we have $\delta_k \approx 0.6$ and the reachable accuracy is $\min_{s \in [0, t]} \|r_k(s)\| \approx 10^{-5}$. In practical implementations of the method a warning should be given if such an accuracy drop takes place during the restart (i.e., if it turns out that $\min_{s \in [0, t]} \|r_k(s)\|$ is larger than the prescribed tolerance tol). We further discuss the RT restarted SAI Krylov subspace methods in Section 3.

3. Numerical experiments

All the numerical tests are carried out on a Linux PC with 8 CPUs Intel Xeon E5504 2.00 GHz in Matlab. No parallelization is carried out for the regular Arnoldi method. However, for Arnoldi/SAI methods automatic parallelization of the sparse LU factorizations is employed by Matlab.

3.1. Convection–diffusion test problem

In this test problem the matrix A is a standard five point central-difference discretization of the following convection–diffusion operator acting on $u(x, y)$, defined for $(x, y) \in [0, 1] \times [0, 1]$ and satisfying the homogeneous Dirichlet boundary

Table 1

Results for the convection–diffusion test problem. The restart ways are denoted by “Res” (residual based), “NH” (Niehoff–Hochbruck), “RT” (residual-time) and “ART” (adaptive RT). EXPOKIT uses time-stepping restarts.

Method	Restart length, way	CPU time (s)	Steps	Error
<i>Pe</i> = 200, mesh 802 × 802				
EXPOKIT	30	57.3	800	3.82e−08
Arnoldi	30, Res	67.8	316	1.18e−07
Arnoldi	30, NH	71.7	317	1.02e−08
Arnoldi	30, RT	44.6	569	2.28e−08
Arnoldi	30, ART	41.1	572	2.05e−08
EXPOKIT	40	63.6	756	5.22e−09
Arnoldi	40, Res	74.9	298	4.40e−08
Arnoldi	40, NH	78.1	299	9.80e−09
Arnoldi	40, RT	45.1	505	1.18e−08
Arnoldi	40, ART	42.9	499	1.27e−08
<i>Pe</i> = 300, mesh 1202 × 1202				
EXPOKIT	30	129.2	800	3.25e−08
Arnoldi	30, Res	136.9	310	8.34e−08
Arnoldi	30, NH	145.0	312	9.99e−09
Arnoldi	30, RT	90.7	539	2.83e−08
Arnoldi	30, ART	85.8	538	2.55e−08
EXPOKIT	40	147.3	756	4.09e−09
Arnoldi	40, Res	161.1	292	2.51e−08
Arnoldi	40, NH	154.2	293	1.08e−08
Arnoldi	40, RT	98.3	489	1.26e−08
Arnoldi	40, ART	93.7	492	1.01e−08

conditions:

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

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

$$v_1(x, y) = x + y, \quad v_2(x, y) = x - y,$$

where the convective terms (i.e., the terms containing the first derivatives of u) are written in this specific form to guarantee that the contribution of the convection terms results in a skew-symmetric matrix [39]. In the experiments, we use matrices A discretized on mesh 802 × 802 for the Peclet number $Pe = 200$ and on mesh 1202 × 1202 for the Peclet number $Pe = 300$. The problem size for these meshes is then $n = 640\,000$ and $n = 1\,440\,000$, respectively. In both cases $\|\frac{1}{2}(A + A^T)\|_2 \approx 6000$ and $\|\frac{1}{2}(A - A^T)\|_2 \approx 0.5$, so that the matrices are weakly non-symmetric. The initial vector v is set to the values of the function $\sin(\pi x)\sin(\pi y)$ on the finite-difference mesh and then normalized as $v := v/\|v\|$. The final time is set to $t = 1$ and the tolerance to $\text{tol} = 10^{-6}$.

We compare the Krylov subspace method based on the Arnoldi process and our RT restarting with the following three methods:

1. The `phiv` function of the EXPOKIT package [10] based on the EXPOKIT time stepping restarting strategy.
2. Krylov subspace method based on the Arnoldi process and residual restarting [13, Chapter 3].
3. Krylov subspace method based on the Arnoldi process and NH (Niehoff–Hochbruck) restarting [13, Chapter 3].

We note that our implementations of the Arnoldi method do not include reorthogonalization of the Krylov basis vectors and we have not noticed a serious orthogonality loss. In the tables of this section presenting numerical results, the last two restarting methods are indicated by “Res” and “NH”, respectively. In these tables, the restarting methods presented in this paper are denoted by “RT” (residual-time) and “ART” (adaptive RT) restarting.

A simple way to restart Krylov subspace evaluations of the matrix exponential is to split the time interval into smaller intervals (time steps), on which the method converges within an acceptable number of Krylov steps. A question arises how this time-stepping restarting approach is compared to our RT restarting. To answer this question we include into comparisons the EXPOKIT package, which exploits such a time-stepping restarting [10].

The results of the comparisons are presented in Tables 1 and 2. The values reported there in the column “error” are relative error norms computed with respect to the EXPOKIT solution. The results in Table 1 show that our RT and ART restarting strategies outperform the other restarting strategies in terms of the CPU time. The residual based and NH restarting perform worse than the EXPOKIT restarting in terms of the CPU time. The reason for this is probably the sophisticated treatment of the projected problems in these restarting strategies, which creates an overhead in

Table 2

Results for the convection–diffusion test problem for the Arnoldi/SAI method. The restart ways are denoted by “Res” (residual based), “NH” (Niehoff–Hochbruck), “RT” (residual-time) and “ART” (adaptive RT). EXPOKIT uses time-stepping restarts.

Method	Restart length, way	CPU time (s)	Steps	Error
<i>Pe</i> = 200, mesh 802×802				
Arnoldi/SAI	30, residual	46.8	16	$1.35\text{e}-09$
Arnoldi/SAI	30, NH	46.2	14	$8.52\text{e}-09$
Arnoldi/SAI	30, RT	44.7	14	$8.52\text{e}-09$
Arnoldi/SAI	10, residual	48.1	18	$3.90\text{e}-07$
Arnoldi/SAI	10, NH	45.4	14	$1.24\text{e}-08$
Arnoldi/SAI	10, RT	46.2	20	$2.50\text{e}-07$
Arnoldi/SAI	5, residual	54.5	20	$1.08\text{e}-09$
Arnoldi/SAI	5, NH	45.4	14	$1.43\text{e}-08$
Arnoldi/SAI	5, RT	43.8	13	$4.25\text{e}-05$
<i>Pe</i> = 300, mesh 1202×1202				
Arnoldi/SAI	10, residual	159.8	17	$2.58\text{e}-07$
Arnoldi/SAI	10, NH	155.5	14	$8.16\text{e}-09$
Arnoldi/SAI	10, RT	154.3	14	$8.03\text{e}-08$
Arnoldi/SAI	5, residual	159.1	15	$1.75\text{e}-07$
Arnoldi/SAI	5, NH	155.5	14	$8.31\text{e}-09$
Arnoldi/SAI	5, RT	153.6	15	$8.14\text{e}-06$
<i>Pe</i> = 300, mesh 1202×1202 , $t = 10$				
Arnoldi/SAI	30, Res	110.3	17	$2.94\text{e}-09$
Arnoldi/SAI	30, NH	108.0	15	$1.25\text{e}-08$
Arnoldi/SAI	30, RT	106.0	15	$1.25\text{e}-08$
Arnoldi/SAI	10, Res	109.5	18	$1.42\text{e}-06$
Arnoldi/SAI	10, NH	106.5	15	$1.57\text{e}-08$
Arnoldi/SAI	10, RT	104.0	15	$1.57\text{e}-05$

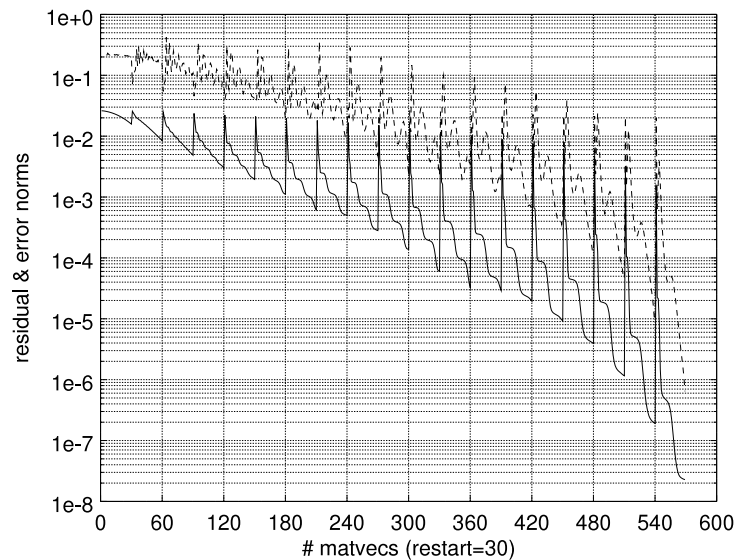


Fig. 8. Residual (dashed line) and error (solid line) convergence plots for the convection–diffusion test problem. $Pe = 200$, mesh 802×802 , the restart length 30.

computational costs. Indeed, if, for instance, 5 restarts of length $k_{\max} = 30$ are carried out with the NH restarting then the matrix exponential of a matrix size 150×150 has to be computed. Additional costs in the residual based restarting is solution of a small projected ODE system, which, for this restarting, cannot be solved by a matrix exponential evaluation [12]. Note also that the matrix–vector products are relatively cheap for this two-dimensional problems, which makes the other costs more pronounced.

In Fig. 8 we present residual and error convergence plots for $Pe = 200$, mesh 802×802 and the restart length 30. We see that the restarts do disturb the convergence which seems unavoidable in general. In Table 2 the results for Arnoldi/SAI method are presented. In this case the method itself converges much quicker than the regular Arnoldi method, hence,

fewer matrix–vector products are required and the restarting effects are less visible. However, we see that all the restarting strategies perform well in this case. The drawback of our RT restarting is that, as discussed in Section 2.5, it tends to deliver a less accurate solution for shorter restart lengths. We plan to address this problem in future.

3.2. Photonic crystal test problem

This test problem is a space-discretized system of the three-dimensional Maxwell equations in a lossless and source-free medium:

$$\begin{aligned}\frac{\partial \mathbf{H}}{\partial t} &= -\frac{1}{\mu} \nabla \times \mathbf{E}, \\ \frac{\partial \mathbf{E}}{\partial t} &= \frac{1}{\varepsilon} \nabla \times \mathbf{H},\end{aligned}\tag{22}$$

where ε and μ are scalar functions of (x, y, z) representing permittivity and permeability, respectively, and \mathbf{H} and \mathbf{E} are vector-valued functions of (x, y, z, t) representing unknown magnetic and electric fields, respectively. The boundary conditions set the tangential electric field components to zero. Physically these conditions can mean perfectly conducting domain boundary or the so-called “far field condition”, see e.g. [40,41], implying that the boundary is placed far enough from the phenomena of interest. For simulations in an unbounded domain with the perfectly matched layer (PML [42,43]) boundary conditions we refer to [44]. The spatial setup for this test problem (along with the boundary conditions) is taken from the second test in [41]: in a spatial domain $[-6.05, 6.05] \times [-6.05, 6.05] \times [-6.05, 6.05]$ filled with air (relative permittivity $\varepsilon_r = 1$) there is a dielectric specimen occupying the region $[-4.55, 4.55] \times [-4.55, 4.55] \times [-4.55, 4.55]$. The specimen has 27 spherical voids ($\varepsilon_r = 1$) of radius 1.4, whose centers have coordinates $(x_i, y_j, z_k) = (3.03i, 3.03j, 3.03k)$, $i, j, k = -1, 0, 1$. The relative permittivity in the dielectric specimen is $\varepsilon_r = 5.0$. The initial values are zero for all the components of both fields \mathbf{H} and \mathbf{E} except for the x - and y -components of \mathbf{E} : they are set to nonzero values in the middle of the spatial domain to represent a light emission. In addition, the initial value vector v is normalized as $v := v/\|v\|$. The standard finite-difference staggered Yee discretization in space leads to an ODE system of the form (2). The meshes $40 \times 40 \times 40$ and $80 \times 80 \times 80$ used in this test lead to problem size $n = 413\,526$ and $n = 3\,188\,646$, respectively. Comparing the results obtained for the two meshes we conclude that this spatial resolution should be sufficient for this test. The final time is now set to $t = 10$ and the tolerance to $\text{tol} = 10^{-6}$.

In the second test problem we compare the two solvers which come out as the best for the first test problem, namely the `phiv` solver of EXPOKIT [10] and our RT restarting (along with its adaptive version ART). Our experience reveals that the SAI strategy is inefficient in this test problem, which can be expected due to a strong nonsymmetry of A . This is in contrast to discretized Maxwell equations with nonreflecting boundary conditions (for instance, to mimic wave propagation in unbounded domains) or in lossy media, where SAI can be efficient [44]. We note that nonzero conductivity terms (lossy media) as well as nonreflecting boundary conditions, such as PML [42,43], lead to a nonzero symmetric part in the matrix A which is beneficial for convergence and overall efficiency of the Krylov subspace exponential time integration [44,45].

The results are presented in Table 3, where the values reported there in the column “error” are relative error norms computed with respect to the EXPOKIT solution. As we see, the RT restarting performs well comparably to EXPOKIT and outperforms it for longer restarts. We also see that the ART restarting, as expected, indeed helps to reduce the CPU time, possibly at the cost of the increased total number of Krylov steps. In Fig. 9 the residual and error convergence plots are presented for the $40 \times 40 \times 40$ mesh and the restart length 30. We observe a convergence stagnation until the last two or three restarts. This is probably caused by the fact that the nonzero entries in the initial vector v are localized at the center of the domain which is not the case for the solution vector $\exp(-tA)v$ for the final time $t = 10$. Since A is a local (finite difference) discretization of a differential operator, the Krylov subspace basis vectors v, Av, A^2v, \dots cannot represent the solution vector well unless the subspace dimension is large enough. Note that there is no stagnation in the convection–diffusion test (cf. Fig. 8) where the initial vector v is not localized.

4. Conclusions and outlook

In this paper a new restarting RT (residual-time) procedure for Krylov subspace matrix exponential evaluations is proposed, analyzed and tested numerically. Our restarting is algorithmically simple as it only relies on evaluation of the readily available residual (8) and the restarted problem has the same form (2) as the original one. Furthermore, the RT restarting compares favorably to three other restarting techniques, namely, the time step restarting of EXPOKIT [10], the generalized residual restarting of Niehoff–Hochbruck [13] and the residual-based restarting [11,12]. For the rational SAI (shift-and-invert) Krylov subspace approximations the proposed restarting works well for moderate accuracy requirements. The RT restarting is also implemented adaptively and the adaptive RT (ART) Krylov subspace method is available as a part of the Octave/Matlab package `expmARPACK` at <http://team.kiam.ru/botchev/expm/>.

Our future research plans include extension of this restarting approach to nonhomogeneous and nonlinear ODE systems, in combination with the exponential block Krylov method [46]. Furthermore, it would be interesting to see how this approach will work for second order ODE systems, where the matrix cosine and sine functions are involved.

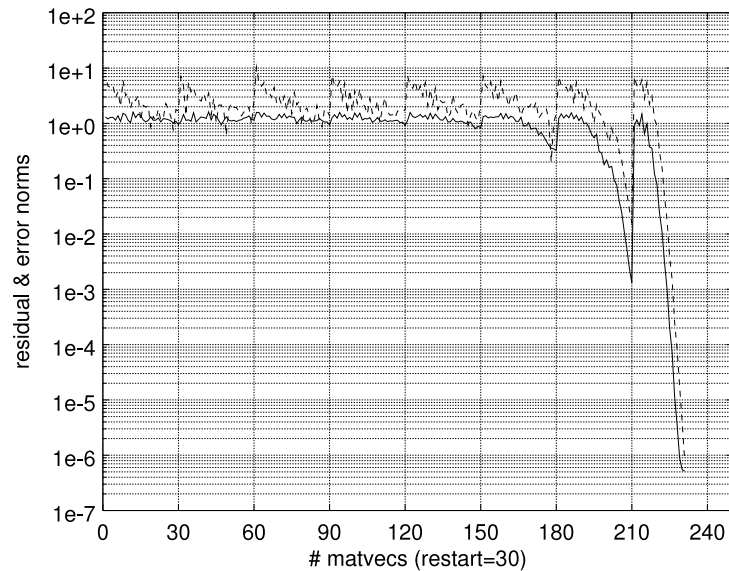


Fig. 9. Residual (dashed line) and error (solid line) convergence plots for the photonic crystal test problem, mesh $40 \times 40 \times 40$, the restart length 30.

Table 3

Results for the photonic crystal test problem. The restart ways are denoted by “RT” (residual-time) and “ART” (adaptive RT). EXPOKIT uses time-stepping restarts.

Method	Restart length, way	CPU time (s)	Steps	Error
Mesh $40 \times 40 \times 40$				
EXPOKIT	30	10.6	256	$1.02\text{e}-06$
Arnoldi	30, RT	11.3	231	$5.19\text{e}-07$
Arnoldi	30, ART	11.2	229	$5.80\text{e}-07$
EXPOKIT	70	23.3	288	$2.16\text{e}-09$
Arnoldi	70, RT	14.4	168	$1.17\text{e}-07$
Arnoldi	70, ART	14.4	167	$1.57\text{e}-07$
Mesh $80 \times 80 \times 80$				
EXPOKIT	30	155.2	512	$4.37\text{e}-07$
Arnoldi	30, RT	162.0	502	$3.15\text{e}-07$
Arnoldi	30, ART	160.8	488	$4.06\text{e}-07$
EXPOKIT	40	170.3	420	$2.25\text{e}-06$
Arnoldi	40, RT	160.8	427	$2.77\text{e}-07$
Arnoldi	40, ART	168.9	417	$9.69\text{e}-07$
EXPOKIT	50	178.9	416	$2.33\text{e}-07$
Arnoldi	50, RT	169.1	383	$1.52\text{e}-07$
Arnoldi	50, ART	169.1	379	$2.31\text{e}-07$
EXPOKIT	60	208.1	434	$2.43\text{e}-08$
Arnoldi	60, RT	181.3	354	$1.06\text{e}-07$
Arnoldi	60, ART	169.0	378	$1.49\text{e}-07$
EXPOKIT	70	250.5	432	$2.74\text{e}-08$
Arnoldi	70, RT	190.1	338	$8.16\text{e}-08$
Arnoldi	70, ART	162.2	389	$1.06\text{e}-07$

A computable expression for the residual, similar to (7),(8), can be derived in this case, too (see e.g. [46]). However, the error and residual behavior in Krylov subspace methods is essentially different, as the projected first order ODE system in (13) is replaced by a corresponding second order counterpart (for more detail see e.g. [47–49]). Thus, it remains to be seen whether the proposed restarting approach can be efficient for the second order ODE systems.

Acknowledgment

The authors thank Vladimir Druskin for stimulating discussions.

References

- [1] I. Moret, P. Novati, An interpolatory approximation of the matrix exponential based on Faber polynomials, *J. Comput. Appl. Math.* 131 (1–2) (2001) 361–380.
- [2] H. Tal-Ezer, Polynomial approximation of functions of matrices and applications, *J. Sci. Comput.* 4 (1) (1989) 25–60.
- [3] H. De Raedt, K. Michielsen, J.S. Kole, M.T. Figge, One-step finite-difference time-domain algorithm to solve the Maxwell equations, *Phys. Rev. E* 67 (2003) 056706.
- [4] B.P. Sommeijer, L.F. Shampine, J.G. Verwer, RKC: An explicit solver for parabolic PDEs, *J. Comput. Appl. Math.* 88 (1998) 315–326.
- [5] V.I. Lebedev, Explicit difference schemes for solving stiff systems of ODEs and PDEs with complex spectrum, *Russian J. Numer. Anal. Math. Modelling* 13 (2) (1998) 107–116.
- [6] M.A. Botchev, G.L.G. Sleijpen, H.A. van der Vorst, Stability control for approximate implicit time stepping schemes with minimum residual iterations, *Appl. Numer. Math.* 31 (3) (1999) 239–253.
- [7] A.H. Al-Mohy, N.J. Higham, Computing the action of the matrix exponential, with an application to exponential integrators, *SIAM J. Sci. Comput.* 33 (2) (2011) 488–511, <http://dx.doi.org/10.1137/100788860>.
- [8] M. Eiermann, O.G. Ernst, A restarted Krylov subspace method for the evaluation of matrix functions, *SIAM J. Numer. Anal.* 44 (2006) 2481–2504.
- [9] H. Tal-Ezer, On restart and error estimation for Krylov approximation of $w = f(A)v$, *SIAM J. Sci. Comput.* 29 (6) (2007) 2426–2441.
- [10] R.B. Sidje, Expokit. A software package for computing matrix exponentials, *ACM Trans. Math. Software* 24 (1) (1998) 130–156, www.maths.uq.edu.au/expokit/.
- [11] E. Celledoni, I. Moret, A Krylov projection method for systems of ODEs, *Appl. Numer. Math.* 24 (2–3) (1997) 365–378.
- [12] M.A. Botchev, V. Grimm, M. Hochbruck, Residual, restarting and Richardson iteration for the matrix exponential, *SIAM J. Sci. Comput.* 35 (3) (2013) A1376–A1397, <http://dx.doi.org/10.1137/110820191>.
- [13] J. Niehoff, Projektionsverfahren Zur Approximation Von Matrixfunktionen Mit Anwendungen Auf Die Implementierung Exponentieller Integratoren (Ph.D. thesis), Mathematisch-Naturwissenschaftlichen Fakultät der Heinrich-Heine-Universität Düsseldorf, 2006.
- [14] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, *SIAM J. Sci. Comput.* 19 (5) (1998) 1552–1574.
- [15] Y. Saad, Analysis of some Krylov subspace approximations to the matrix exponential operator, *SIAM J. Numer. Anal.* 29 (1) (1992) 209–228.
- [16] M. Afanasjew, M. Eiermann, O.G. Ernst, S. Güttel, Implementation of a restarted Krylov subspace method for the evaluation of matrix functions, *Linear Algebra Appl.* 429 (2008) 2293–2314.
- [17] S. Güttel, Rational Krylov Methods for Operator Functions (Ph.D. thesis), Technischen Universität Bergakademie Freiberg, 2010, www.guettel.com.
- [18] M. Eiermann, O.G. Ernst, S. Güttel, Deflated restarting for matrix functions, *SIAM J. Matrix Anal. Appl.* 32 (2) (2011) 621–641.
- [19] S. Güttel, A. Frommer, M. Schweitzer, Efficient and stable Arnoldi restarts for matrix functions based on quadrature, *SIAM J. Matrix Anal. Appl.* 35 (2) (2014) 661–683.
- [20] T. Jawecki, W. Auzinger, O. Koch, Computable strict upper bounds for Krylov approximations to a class of matrix exponentials and ϕ -functions, arXiv preprint [arXiv:1809.03369](https://arxiv.org/abs/1809.03369), 2018. <https://arxiv.org/pdf/1809.03369>.
- [21] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, Springer Verlag, 2003.
- [22] K. Dekker, J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Non-Linear Differential Equations, Elsevier Science Publishers, North-Holland, 1984.
- [23] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore and London, 1996.
- [24] B.N. Parlett, The Symmetric Eigenvalue Problem, SIAM, 1998.
- [25] H.A. van der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge University Press, 2003.
- [26] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, 2003, Available from <http://www-users.cs.umn.edu/saad/books.html>.
- [27] H.A. van der Vorst, An iterative solution method for solving $f(A)x = b$, using krylov subspace information obtained for the symmetric positive definite matrix A , *J. Comput. Appl. Math.* 18 (1987) 249–263.
- [28] V.L. Druskin, L.A. Knizhnerman, Two polynomial methods of calculating functions of symmetric matrices, *U.S.S.R. Comput. Maths Math. Phys.* 29 (6) (1989) 112–121.
- [29] L.A. Knizhnerman, Calculation of functions of unsymmetric matrices using Arnoldi's method, *U.S.S.R. Comput. Maths Math. Phys.* 31 (1) (1991) 1–9.
- [30] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential operator, *SIAM J. Numer. Anal.* 34 (5) (1997) 1911–1925.
- [31] V.L. Druskin, A. Greenbaum, L.A. Knizhnerman, Using nonorthogonal Lanczos vectors in the computation of matrix functions, *SIAM J. Sci. Comput.* 19 (1) (1998) 38–54.
- [32] B. Beckermann, L. Reichel, Error estimation and evaluation of matrix functions via the Faber transform, *SIAM J. Num. Anal.* 47 (2009) 3849–3883.
- [33] P.K. Suetin, Series of Faber Polynomials, CRC Press, 1998.
- [34] B. Beckermann, Image numérique, GMRES et polynômes de Faber, *C. R. Acad. Sci. Paris I* 340 (11) (2005) 855–860.
- [35] M. Hochbruck, A. Ostermann, Exponential integrators, *Acta Numer.* 19 (2010) 209–286.
- [36] I. Moret, P. Novati, RD rational approximations of the matrix exponential, *BIT* 44 (2004) 595–615.
- [37] J. van den Eshof, M. Hochbruck, Preconditioning Lanczos approximations to the matrix exponential, *SIAM J. Sci. Comput.* 27 (4) (2006) 1438–1457.
- [38] T. Gökler, V. Grimm, Uniform approximation of ϕ -functions in exponential integrators by a rational Krylov subspace method with simple poles, *SIAM J. Matrix Anal. Appl.* 35 (4) (2014) 1467–1489, <http://dx.doi.org/10.1137/140964655>.
- [39] 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. Tz* 7 (7) (1979) 43–55, Translation from *Izv. Vyssh. Uchebn. Zaved., Mat.* 1979, No. 7(206), 41–52 (1979).
- [40] A. Taflov, S.C. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, third ed., Artech House Inc, Boston, MA, 2005.
- [41] J.S. Kole, M.T. Figge, H. De Raedt, Unconditionally stable algorithms to solve the time-dependent Maxwell equations, *Phys. Rev. E* 64 (2001) 066705.
- [42] J. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, *J. Comput. Phys.* 114 (1994) 185–200.
- [43] S.G. Johnson, Notes on perfectly matched layers (PMLs). [math.mit.edu/stevenj/18.369/pml.pdf](http://math.mit.edu/~stevenj/18.369/pml.pdf), 2010.
- [44] M.A. Botchev, Krylov subspace exponential time domain solution of Maxwell's equations in photonic crystal modeling, *J. Comput. Appl. Math.* 293 (2016) 24–30, <http://dx.doi.org/10.1016/j.cam.2015.04.022>.
- [45] J.G. Verwer, M.A. Botchev, Unconditionally stable integration of Maxwell's equations, *Linear Algebra Appl.* 431 (3–4) (2009) 300–317.
- [46] M.A. Botchev, A block Krylov subspace time-exact solution method for linear ordinary differential equation systems, *Numer. Linear Algebra Appl.* 20 (4) (2013) 557–574.
- [47] M. Hochbruck, C. Lubich, A gautschi-type method for oscillatory second-order differential equations, *Numer. Math.* 83 (1999) 403–426.
- [48] M.A. Botchev, D. Harutyunyan, J.J.W. van der Vegt, The Gautschi time stepping scheme for edge finite element discretizations of the Maxwell equations, *J. Comput. Phys.* 216 (2006) 654–686, <http://dx.doi.org/10.1016/j.jcp.2006.01.014>.
- [49] V. Grimm, M. Hochbruck, Rational approximation to trigonometric operators, *BIT* 48 (2) (2008) 215–229.