Solving anisotropic heat equations by exponential shift-and-invert and polynomial Krylov subspace methods

M. A. Botchev

Keldysh Institute of Applied Mathematics, RAS, Moscow
E-mail: botchev@ya.ru

Abstract. We assess performance of the exponential Krylov subspace methods for solving a class of parabolic problems with a strong anisotropy in coefficients. Different boundary conditions are considered, which have a direct impact on the smallest eigenvalue of the discretized operator and, hence, on the convergence behavior of the exponential Krylov subspace solvers. Restarted polynomial Krylov subspace methods and shift-and-invert Krylov subspace methods combined with algebraic multigrid are considered.

1. Introduction

Exponential time integration schemes have been proved to be an efficient tool in real life applications [1, 2, 3] and form an actively developing research scene [4]. These schemes have a potential to produce accurate and stable solutions to various time-dependent problems and have been shown to be efficient especially for parabolic partial differential equations (PDEs). Being exact for typical model linear problems, exponential schemes also possess excellent stability properties. Exponential time integration schemes are characterized by the matrix exponential (for definition see, e.g., [5, 6, 7]) or other related matrix functions (such as, for hyperbolic problems, the matrix sine and cosine) present in their formulation. For parabolic problems, besides the matrix exponential, these schemes may include the φ matrix function defined as

\[ \varphi(z) \equiv \frac{e^z - 1}{z}, \]

where we set, by definition, \( \varphi(0) \equiv 1 \). To implement actions of these matrix functions on a vector in large scale problems, different numerical linear algebra techniques exist, including ones based on the Chebyshev polynomials, scaling and squaring with Padé or Taylor approximations, real Leja points method and other, see, e.g., [8, 9, 10, 11, 12]. A versatile and efficient approach for evaluating the matrix exponential and φ matrix functions are the Krylov subspace methods [13, 14], with some early works on matrix functions including [15, 16, 17, 18, 19, 20]. Although the exponential schemes have been recognized as a promising tool in solving some large scale problems [21, 22], their efficiency for specific application areas often remains to be proved. This is, in particular, the case for applications based on parabolic PDEs, where numerous efficient time integration approaches have been developed including splitting, linearized Runge-Kutta schemes, explicit iterative schemes and many other, see, e.g., [23]. There
are several reasons why in real life computations these conventional time integration schemes can be more efficient than at first sight very promising exponential schemes. First, exponential time integration schemes may tend to produce too accurate and, hence, potentially too expensive, in terms of computational costs, results [24]. Recall that for PDE problems the total numerical error is typically dominated by the space discretization error [23], so that a moderate accuracy in time often suffices. Second, modern implementations of conventional time integration schemes often rely on efficient numerical linear techniques such Krylov subspace methods [25, 26] or highly parallel Chebyshev and multigrid iterations [27, 28]. Hence, a conventional implicit time integration scheme based, for instance, on an optimized Krylov subspace linear system solver, can be quite similar, at least from a computational point of view, to an exponential Krylov subspace time integrator.

The aim of this paper is to thoroughly assess the performance of the Krylov subspace exponential time integrators for a particular class of anisotropic parabolic problems (given below by (2)). The paper is organized as follows. In Section 2 we present the Krylov subspace methods being evaluated in this paper. In Section 3 numerical tests are described and their results are discussed. Some conclusions are drawn in the last section.

2. The test problems and methods

Unless indicated otherwise, throughout the paper || · || denotes the 2-norm.

2.1. The test problem

For \((x, y, z) \in \Omega \equiv [0, 1] \times [0, 1] \times [0, 1]\) we consider initial-boundary value problem

\[ u_t = k_1 u_{xx} + k_2 u_{yy} + k_3 u_{zz}, \quad u|_{t=0} = u^0(x, y, z), \quad (2) \]

where \(u^0(x, y, z)\) is a given function and one of the three following boundary conditions are imposed:

- either \(u|_{\partial \Omega} = 0\), \(u|_{\partial \Omega} \neq 0\), \(\partial u \bigg|_{\partial \Omega} = 0\), \(\partial u \bigg|_{x=0,x=1,y=0,y=1} = 0\), \(u|_{z=0,z=1} = 0\) (3)

where \(\partial u / \partial n\) denotes the derivative of \(u\) in the normal direction. It is not difficult to see that any analytic solution of (2) can be represented as a possibly infinite sum of the eigenmodes, i.e.,

\[ \sum_{i,j,\ell} e^{-\lambda_{ij\ell} t} u_{ij\ell}(x, y, z), \quad \lambda_{ij\ell} = \pi^2 (k_1 i^2 + k_2 j^2 + k_3 \ell^2), \quad (6) \]

with summation ranges and \(u_{ij\ell}(x, y, z)\) defined, depending on the boundary conditions, in Table 1.

Following the method of lines approach, we first apply a space discretization in space. The standard second order finite difference space discretization of the operator \(-k_1 u_{xx} - k_2 u_{yy} - k_3 u_{zz}\) results in a semidiscrete initial-value problem

\[ y'(t) = -Ay(t), \quad y(0) = y^0, \quad (7) \]

where \(A = A^T \in \mathbb{R}^{N \times N}\) is positive (semi)definite matrix and the entries of \(y^0\) are the values of the initial value function \(u^0(x, y, z)\) on the mesh. We note that boundary conditions (3) lead to a nonsingular matrix \(A\), boundary conditions (4) to a singular \(A\), and boundary conditions (5) to a matrix close to singular (cf. (3)). This has direct implications for convergence of the Krylov methods (see, e.g., [29, 24]).
2.2. Krylov subspace exponential time integrators

2.2.1. Krylov subspace methods

Exact solution of the semidiscrete problem \( y(t) = \exp(-tA)y_0 \), where \( \exp(-tA) \) is the matrix exponential. Polynomial Krylov subspace methods compute the matrix-vector product \( \exp(-tA)y_0 \) approximately as follows. First, \( k \) steps \((k \ll N)\) of the Lanczos process (or, for \( A \neq A^T \), Arnoldi process) are carried out. Starting with vector \( v_1 = y_0/\|y_0\| \), this process computes successively the vectors \( v_{q+1}, q = 1, \ldots, k \), by first setting \( v_{q+1} := Av_q \), then, orthogonalizing \( v_{q+1} \) with respect to the last two computed vectors \( v_{q-1}, v_q \) and then normalizing \( v_{q+1} \), see, e.g., \[13\] \[14\] for details. This yields a matrix \( V_{k+1} = \mathbb{R}^{N \times (k+1)} \), with the orthonormal columns \( v_1, \ldots, v_{k+1} \), and a tridiagonal matrix \( H_{k+1,k} = \mathbb{R}^{(k+1) \times k} \) such that

\[
V_k = V_{k+1} H_{k+1,k} = V_k H_{k,k} + h_{k+1,k} e_k^T v_{k+1},
\]

where \( H_{k,k} \in \mathbb{R}^{k \times k} \) is formed by the first \( k \) rows of \( H_{k+1,k} \), \( h_{k+1,k} \) is the only nonzero entry in the last row of \( H_{k+1,k} \), and \( e_k = (0, \ldots, 0, 1)^T \in \mathbb{R}^k \). Then the Krylov subspace approximation \( y_k(t) \) to \( \exp(-tA)y_0 \) is computed as

\[
y(t) = \exp(-tA)y_0 = \beta \exp(-tA)V_ke_1 \approx y_k(t) = \beta V_k \exp(-tH_{k,k})e_1,
\]

where \( \beta = \|y_0\|, e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^k \). We emphasize that the matrix \( \exp(-tA) \) is not computed and, as we see, to evaluate its product with the vector \( y_0 \), only the matrix exponential \( \exp(-tH_{k,k}) \) of the small matrix \( H_{k,k} \) has to be computed. This is done by standard numerical linear algebra techniques \[6\] \[7\]. Note also that the same matrices \( V_k \) and \( H_{k,k} \) are used for all time moments \( t > 0 \), though the quality of the approximation \( y(t) \approx y_k(t) \) deteriorates with growing \( t \). To know when to stop the Lanczos process, we estimate the norm of the exponential residual \( r_k(t) \), defined as \( r_k(t) = -Ay_k - y_k(t) \), see \[30\] \[31\] \[32\]. Using relations \[8\] \[9\] it is not difficult to see that \[30\] \[31\]

\[
r_k(t) = [-\beta h_{k+1,k} e_k^T \exp(-tH_{k,k})e_1] v_{k+1}, \quad \|r_k(t)\| = \beta h_{k+1,k} |e_k^T \exp(-tH_{k,k})e_1|.
\]

2.2.2. SAI Krylov subspace methods

The conventional polynomial Krylov subspace method discussed above tends to approximate the solution components associated with the large eigenvalues of \( A \) better than the other components \[13\] \[14\]. Therefore, for stiff problems, where the eigenvalues of \( A \) significantly vary in magnitude, it may take many Krylov steps \( k \) before the important low-frequency modes associated with the small eigenvalues of \( A \) are well approximated. To avoid this convergence stagnation, two approaches are typically applied. First, restarting methods are employed which try to preserve the convergence while limiting \( k \), see, e.g., \[33\]. Second, besides conventional polynomial Krylov subspace methods, rational Krylov subspace methods are used \[34\]. Among the rational Krylov subspace methods, the so-called shift-and-invert (SAI) Krylov subspace methods are popular \[35\] \[36\]. They build the Krylov subspace for the shifted-and-inverted matrix \( \tilde{A} = (I + \gamma A)^{-1} \), where usually \( \gamma := t_{\max}/10 \), with \( t_{\max} \) being the time interval length \[36\]. The Lanczos/SAI method produces the matrix \( V_{k+1} \in \mathbb{R}^{N \times (k+1)} \),
with orthonormal columns \( v_1 = y^0 / \| y^0 \|, \ldots, v_{k+1} \), and a matrix \( \tilde{H}_{k+1,k} \in \mathbb{R}^{(k+1) \times k} \) such that a relation similar to (8) holds:

\[
\tilde{A}V_k = V_{k+1}\tilde{H}_{k+1,k} = V_k\tilde{H}_{k,k} + \tilde{h}_{k+1,k}e_k^Tv_{k+1},
\]

where \( \tilde{H}_{k,k} \) and \( \tilde{h}_{k+1,k} \) are defined as in (8). After that the Lanczos/SAI approximation \( y_k(t) \) to \( \exp(-tA)y^0 \) is computed as [35, 36]

\[
H_{k,k} := \frac{1}{\gamma}(\tilde{H}_{k,k} - I), \quad y_k(t) = \beta V_k \exp(-tH_{k,k})e_1,
\]

where \( \beta = \|y_0\|, e_1 = (1,0,\ldots,0)^T \in \mathbb{R}^k \). We emphasize that the SAI matrix \( \tilde{A} = (I + \gamma A)^{-1} \) is not computed. Instead, linear systems with \( I + \gamma A \) are solved every time the matrix \( \tilde{A} \) is multiplied with a vector. These systems do not have to be solved very accurately, an efficient strategy to choose a proper relaxed tolerance for solving these systems exists [36]. To control convergence in the Lanczos/SAI method we monitor the residual norm \( \|r_k(t)\| \), with \( r_k(t) \equiv -Ay_k - y_k'(t) \), which can be computed as [36, 32]

\[
r_k(t) = \frac{\beta}{\gamma}h_{k+1,k}e_k^T\tilde{H}_{k,k}^{-1}\exp(-tH_{k,k})e_1(I + \gamma A)v_{k+1},
\]

\[
\|r_k(t)\| = \frac{\beta}{\gamma}h_{k+1,k}e_k^T\tilde{H}_{k,k}^{-1}\exp(-tH_{k,k})e_1\|I + \gamma A\|v_{k+1}.
\]

2.2.3. Krylov subspace methods for the \( \varphi \) function

Krylov subspace evaluation of the \( \varphi \) matrix function [1] allows to solve exactly a more general problem:

\[
y'(t) = -Ay(t) + g, \quad y(0) = y^0,
\]

where \( g \in \mathbb{R}^N \) is a given vector. It easy to check that exact solution of (14) is

\[
y(t) = y^0 + t\varphi(-tA)(g - Ay^0), \quad t \geq 0.
\]

A Krylov subspace approximation \( y_k(t) \) to \( y(t) \) can be computed in the same way as for the matrix exponential. The Lanczos or Arnoldi process then starts with the vector \( v_1 := (g - Ay^0)/\beta, \beta = \|g - Ay^0\| \) and we have

\[
y_k(t) = y^0 + \beta tV_k\varphi(-tH_{k,k})e_1.
\]

To monitor convergence, similarly to the matrix exponential case, we can check the residual norm \( \|r_k(t)\| \) for the residual \( r_k(t) \) defined as \( r_k(t) \equiv -Ay_k(t) + g - y_k'(t) \) [24]. The SAI approach can be employed here in the same way as for the matrix exponential.

3. Numerical experiments

In all the experiments we set

\[ k_1 = 10^4, \quad k_2 = 10^2, \quad k_3 = 1. \]

The test results are presented for three exponential Krylov subspace solvers:

(i) Lanczos phiRT(30), the Lanczos method for the \( \varphi \) function with residual-time (RT) restarting [24] and maximal Krylov subspace dimension 30, see [8], [16],

(ii) Lanczos/SAI exp, the Lanczos/SAI method for the matrix exponential, see [11], [12], and
Table 2. Convergence of the relative residual norm \(17\) for the dominant analytic eigenvalue \(\lambda_{1,1,1}\) and the corresponding eigenvector, for different boundary conditions (b.c.)

| mesh     | b.c. (3) | b.c. (4) | b.c. (5) |
|----------|----------|----------|----------|
| 10 \times 12 \times 14 | 6.739e+02 | 8.147e+02 | 8.147e+02 |
| 20 \times 24 \times 28 | 1.852e+02 | 2.042e+02 | 2.042e+02 |
| 40 \times 48 \times 56 | 4.862e+01 | 5.108e+01 | 5.108e+01 |
| 80 \times 96 \times 112 | 1.246e+01 | 1.277e+01 | 1.277e+01 |
| 160 \times 192 \times 224 | 3.154e+00 | 3.193e+00 | 3.193e+00 |

Table 3. Results for the test problem \(2\) with Dirichlet boundary conditions \(3\)

| solver            | relative error \(18\) | Krylov steps/ CPU time, s |
|-------------------|------------------------|---------------------------|
| 20 \times 22 \times 24 mesh |                        |                           |
| Lanczos phiRT(30) | 2.132e-02              | 234 / 0.68                |
| Lanczos/SAI exp   | 2.132e-02              | 16 / 0.64                 |
| Lanczos/SAI phi   | 2.132e-02              | 10 / 0.61                 |
| 40 \times 44 \times 48 mesh |                        |                           |
| Lanczos phiRT(30) | 5.550e-03              | 805 / 5.9                 |
| Lanczos/SAI exp   | 5.550e-03              | 16 / 20.3                 |
| Lanczos/SAI phi   | 5.550e-03              | 10 / 19.3                 |
| 80 \times 88 \times 96 mesh |                        |                           |
| Lanczos phiRT(30) | 1.419e-03              | 2722 / 109                |
| Lanczos/SAI exp   | 1.419e-03              | 16 / 1322                 |
| Lanczos/SAI phi   | 1.419e-03              | 10 / 1297                 |

(iii) Lanczos/SAI phi, the Lanczos/SAI method for the \(\varphi\) function, see \(11\), \(19\).

In the SAI solvers we set \(\gamma\) to the time interval length divided by 10.

The tests described in this section are carried out in MATLAB on a Linux cluster. If not reported otherwise, the linear systems with the matrix \(I + \gamma A\) in the SAI methods are solved by the standard sparse direct MATLAB solvers, which employ UMFPACK \(37\).

Table 2 shows, for the matrices \(A\) computed on different meshes and for different boundary conditions, the relative residual norm

\[
\|Av_{1,1,1} - \lambda_{1,1,1}v_{1,1,1}\|/\|v_{1,1,1}\|, \quad (17)
\]

where the vector \(v_{1,1,1}\) contains the mesh values of the analytic eigenmode \(u_{1,1,1}(x,y,z)\) on the mesh, see relation \(6\) and Table 1. A second order convergence can be seen for all boundary conditions, which confirms that the matrices \(A\) are computed correctly.

In all the tests the final time is set to \(t = 10^{-4}\). The relative error values reported below for the exponential solvers are computed as

\[
\|y_k(t) - y_{\text{exact}}(t)\|/\|y_{\text{exact}}(t)\|, \quad (18)
\]

where \(y_k(t)\) is numerical solution and \(y_{\text{exact}}(t)\) contains the mesh values of the analytical solution for the final time \(t\). Since the reference solution is the analytical solution, the accuracy that can be achieved is restricted by the spatial error. For this error a second order convergence can be expected as the grid is refined.
All the exponential solvers use the residual stopping criterion with a stringent tolerance $\texttt{tol} = 1e-10$. This means that each of the solvers stops as soon as for all $t \in [0, 10^{-4}]$

$$\|r_k(t)\|/\beta \leq \texttt{tol},$$

where $\beta$ is either $\|g^0\|$ (for the Lanczos/SAI $\texttt{exp}$ solver) or $\|g - Ay^0\|$ (for the other two solvers). Taking into account that the error being measured is restricted by the spatial error, the chosen tolerance can be seen as too stringent in the sense that a similar total accuracy can probably be achieved for a much milder stopping criterion for $\|r_k(t)\|$. We choose this stringent tolerance value to check convergence properties of the exponential solvers.

In Table 3 we show results for numerical solution of problem (2),(3) where $y^0$ is chosen such that the analytical solution is $u_{\text{exact}}(x, y, z, t) = \sum_{i,j,\ell}^{3} e^{-\lambda_{ij\ell} t} u_{ij\ell}(x, y, z)$, with $\lambda_{ij\ell}$ and $u_{ij\ell}(x, y, z)$ defined respectively in (6) and Table 1. For this problem and the chosen final time $t = 10^{-4}$ we have $\|A\| > 26 000$ on the mesh $80 \times 88 \times 96$, which gives a rough indication of the number of time steps needed by an explicit scheme. As expected, a second order convergence of the relative error can be seen as the grid gets refined.

In Table 3 we also see that the Lanczos/SAI $\texttt{phi}$ solver always requires a less number of Krylov steps than the Lanczos/SAI $\texttt{exp}$ solver. This is due to the difference in the initial vector for the Lanczos process (see relation (19)) and not due a different convergence rate. In fact, both solvers exhibit roughly the same convergence rate, as can be seen in Figure 1. We also note that the tolerance $\texttt{tol} = 1e-10$ is indeed too stringent: similar relative error values can be achieved for the residual tolerance $\texttt{tol}$ set in (19) to $1e-5$ or $1e-6$.

Table 3 shows that the Lanczos/SAI solvers require the same number of Krylov steps for all the meshes, while its CPU time grows significantly as the grid gets finer. This is due to the increasing costs for computing the actions of $(I + \gamma A)^{-1}$ done by the sparse direct solver.

To test the solver performance for problem (2),(4), we set the initial value vector $y^0$ to the grid values of an analytic solution taken as (cf. (6)) $u_{\text{exact}}(x, y, z, t) = \sum_{i,j,\ell}^{3} e^{-\lambda_{ij\ell} t} u_{ij\ell}(x, y, z)$. For this problem and the final time $t = 10^{-4}$ we have $\|A\| > 25 000$ on the mesh $80 \times 88 \times 96$. The results are presented in Table 4. We again see the expected second order convergence of the numerical solution and a mesh independent convergence of the SAI Krylov solvers.

In Table 4 we also show results for the exact solution taken to be $u_{\text{exact}}(x, y, z, t) = \sum_{i,j,\ell}^{3} e^{-\lambda_{ij\ell} t} u_{ij\ell}(x, y, z, t)$ (these results are indicated in the table as “$\lambda_{ij\ell} > 0$”). In this case
Table 4. Results for the test problem (2) with Neumann boundary conditions (4)

| solver              | relative error (18) | Krylov steps/ | CPU time, s |
|---------------------|---------------------|---------------|-------------|
| 20 × 22 × 24 mesh   |                     |               |             |
| Lanczos phiRT(30)   | 3.774e-03           | 344 / 1.0     |             |
| Lanczos/SAI exp     | 3.774e-03           | 24 / 0.9      |             |
| Lanczos/SAI phi     | 3.774e-03           | 15 / 0.8      |             |
| 40 × 44 × 48 mesh   |                     |               |             |
| Lanczos phiRT(30)   | 9.432e-04           | 1009 / 7.4    |             |
| Lanczos/SAI exp     | 9.432e-04           | 25 / 22.2     |             |
| Lanczos/SAI phi     | 9.432e-04           | 15 / 19.8     |             |
| Lanczos/SAI/AGMG exp| 9.432e-04           | 28 (433) / 6.6|             |
| Lanczos/SAI/AGMG phi| 9.432e-04           | 15 (198) / 3.2|             |
| 80 × 88 × 96 mesh   |                     |               |             |
| Lanczos phiRT(30)   | 2.358e-04           | 3466 / 141    |             |
| Lanczos/SAI exp     | 2.358e-04           | 26 / 1241     |             |
| Lanczos/SAI phi     | 2.358e-04           | 15 / 1210     |             |
| Lanczos/SAI/AGMG exp| 2.358e-04           | 30 (518) / 65 |             |
| Lanczos/SAI/AGMG phi| 2.358e-04           | 15 (240) / 31 |             |
| 20 × 22 × 24 mesh, \(\lambda_{ij} > 0\) | | | |
| Lanczos phiRT(30)   | 1.846e-03           | 210 / 0.7     |             |
| Lanczos/SAI exp     | 1.846e-03           | 17 / 0.6      |             |
| Lanczos/SAI phi     | 1.846e-03           | 10 / 0.6      |             |
| 40 × 44 × 48 mesh, \(\lambda_{ij} > 0\) | | | |
| Lanczos phiRT(30)   | 5.828e-03           | 749 / 5.8     |             |
| Lanczos/SAI exp     | 5.828e-03           | 17 / 20.4     |             |
| Lanczos/SAI phi     | 5.828e-03           | 10 / 19.6     |             |
| Lanczos/SAI/AGMG exp| 5.828e-03           | 23 (295) / 4.6|             |
| Lanczos/SAI/AGMG phi| 5.828e-03           | 10 (137) / 2.4|             |
| 80 × 88 × 96 mesh, \(\lambda_{ij} > 0\) | | | |
| Lanczos phiRT(30)   | 1.454e-03           | 2706 / 110    |             |
| Lanczos/SAI exp     | 1.454e-03           | 17 / 1262     |             |
| Lanczos/SAI phi     | 1.454e-03           | 10 / 1175     |             |
| Lanczos/SAI/AGMG exp| 1.454e-03           | 22 (337) / 43.3|         |
| Lanczos/SAI/AGMG phi| 1.454e-03           | 10 (157) / 20.9|         |

the eigenmode corresponding to the zero eigenvalue is set off, which means a better separation of the active eigenmodes from the origin and a faster convergence (see, e.g., [29, 24]). Indeed, as we see, switching these eigenmodes off results in a convergence similar to the reported in Table 3.

Table 4 also includes results for the variants of the SAI solvers where an iterative solver is used to solve the linear systems with the matrix \(I + \gamma A\). As the iterative solver the AGMG (AGgregation-based algebraic MultiGrid) code is employed [38, 39, 40, 41]. We see that the number of the Krylov steps in the exponential SAI methods combined with the AGMG solver are the same or slightly larger than when the SAI methods use the sparse direct solver. A slight increase in the number of outer iterations is due to the relaxed tolerance strategy used by the SAI solvers [36]. Nevertheless, the SAI/AGMG solvers still exhibit a mesh independent convergence and the number of AGMG iterations grows moderately with the grid size.
Table 5. Results for the test problem (2) with Neumann–Dirichlet boundary conditions (5)

| solver                        | relative error (18) | Krylov steps/ CPU time, s |
|-------------------------------|---------------------|---------------------------|
| 20 × 22 × 24 mesh             |                     |                           |
| Lanczos phiRT(30)             | 3.777e-03           | 294 / 0.9                 |
| Lanczos/SAI exp               | 3.777e-03           | 24 / 0.7                  |
| Lanczos/SAI phi               | 3.777e-03           | 15 / 0.6                  |
| Lanczos/SAI/AGMG exp          | 3.777e-03           | 28 (351) / 1.1            |
| Lanczos/SAI/AGMG phi          | 3.777e-03           | 15 (168) / 0.5            |
| 40 × 44 × 48 mesh             |                     |                           |
| Lanczos phiRT(30)             | 9.441e-04           | 1008 / 7.7                |
| Lanczos/SAI exp               | 9.441e-04           | 24 / 21.6                 |
| Lanczos/SAI phi               | 9.441e-04           | 15 / 19.8                 |
| Lanczos/SAI/AGMG exp          | 9.441e-04           | 28 (433) / 6.6            |
| Lanczos/SAI/AGMG phi          | 9.441e-04           | 15 (197) / 3.1            |
| 80 × 88 × 96 mesh             |                     |                           |
| Lanczos phiRT(30)             | 2.360e-04           | 3466 / 142                |
| Lanczos/SAI exp               | 2.360e-04           | 26 / 1288                 |
| Lanczos/SAI phi               | 2.360e-04           | 15 / 1202                 |
| Lanczos/SAI/AGMG exp          | 2.360e-04           | 30 (518) / 65             |
| Lanczos/SAI/AGMG phi          | 2.360e-04           | 15 (240) / 31             |
| 20 × 22 × 24 mesh, \(\lambda_{ij\ell} > 0\) |               |                           |
| Lanczos phiRT(30)             | 2.350e-02           | 209 / 0.6                 |
| Lanczos/SAI exp               | 2.350e-02           | 17 / 0.7                  |
| Lanczos/SAI phi               | 2.350e-02           | 10 / 0.6                  |
| Lanczos/SAI/AGMG exp          | 2.350e-02           | 22 (239) / 0.8            |
| Lanczos/SAI/AGMG phi          | 2.350e-02           | 10 (119) / 0.4            |
| 40 × 44 × 48 mesh, \(\lambda_{ij\ell} > 0\) |               |                           |
| Lanczos phiRT(30)             | 5.828e-03           | 749 / 5.5                 |
| Lanczos/SAI exp               | 5.828e-03           | 17 / 4.5                  |
| Lanczos/SAI phi               | 5.828e-03           | 10 / 2.2                  |
| Lanczos/SAI/AGMG exp          | 5.828e-03           | 23 (295) / 4.5            |
| Lanczos/SAI/AGMG phi          | 5.828e-03           | 10 (137) / 2.2            |
| 80 × 88 × 96 mesh, \(\lambda_{ij\ell} > 0\) |               |                           |
| Lanczos phiRT(30)             | 1.454e-03           | 2636 / 109                |
| Lanczos/SAI exp               | 1.454e-03           | 17 / 1240                 |
| Lanczos/SAI phi               | 1.454e-03           | 10 / 1182                 |
| Lanczos/SAI/AGMG exp          | 1.454e-03           | 22 (336) / 43             |
| Lanczos/SAI/AGMG phi          | 1.454e-03           | 10 (157) / 21             |
| 120 × 132 × 144 mesh, \(\lambda_{ij\ell} > 0\) |               |                           |
| Lanczos phiRT(30)             | 6.460e-04           | 5208 / 688                |
| Lanczos/SAI/AGMG exp          | 6.460e-04           | 23 (361) / 155            |
| Lanczos/SAI/AGMG phi          | 6.460e-04           | 10 (168) / 74             |
The results for problem (2,5) are shown in Table 5. The analytic reference solution for this problem is taken to be (cf. (6))
\[ u_{\text{exact}}(x,y,z,t) = \sum_{i,j=0} e^{-\lambda_{ij}t} u_{ij}(x,y,z). \]
Taking into account the values of \( k_{1,2,3} \), see (3), the smallest eigenvalue of \( A \) is very close to zero in this case. Results indicated in the table by "\( \lambda_{ij} > 0 \)" are obtained for the analytic solution where \( i,j = 1,2,3 \), i.e., with the lowest eigenmode switched off. Table 5 allows to make conclusions similar to those made for Table 4.

We note that on the \( 64 \times 64 \times 64 \) grid the Lanczos phiRT(30) method can reach the best achievable reference error \( 2.342e-03 \) (this value is, as discussed above, determined by the spatial error) within about 900 Krylov steps. To reach the same error, the LI-M (local iterative) stabilized explicit scheme [42] requires about 3500 matrix-vector multiplications. For a comparison between a Krylov subspace SAI solver and the implicit Euler scheme see, e.g., [43, Figure 2].

4. Conclusions
The tested exponential Krylov subspace methods with the residual stopping criterion appear to be a reliable tool for solving parabolic time-dependent problems with strong anisotropy. Tolerance values for the residual which are one or two orders smaller than the spatial error should be taken. The methods demonstrate a mesh independent convergence and a robust behavior with respect to different boundary conditions which have a direct impact on the smallest eigenvalue of the discretized spatial differential operator. Finally, the SAI exponential methods can be efficiently implemented based on algebraic multigrid linear solvers. On the finest \( 120 \times 132 \times 144 \) mesh, the Lanczos/SAI phi method combined with the AGMG algebraic multigrid solver delivers a solution within 74 s of CPU time on a Linux cluster.

Acknowledgments
The author would like to thank Victor Timofeevich Zhukov for useful discussions. A part of the computations reported here are performed on the hybrid supercomputer K-100 installed in the Supercomputer Center of Collective Usage at the Keldysh Institute of Applied Mathematics, RAS.

References
[1] M. Hochbruck, T. Pažur, A. Schulz, Th. Thawinan, and C. Wiener, “Efficient time integration for discontinuous Galerkin approximations of linear wave equations,” ZAMM, vol. 95, no. 3, pp. 237–259, 2015.
[2] X. Fu, C. Wang, P. Li, and L. Wang, “Exponential integration algorithm for large-scale wind farm simulation with Krylov subspace acceleration,” Applied Energy, vol. 254, p. 113692, 2019. https://doi.org/10.1016/j.apenergy.2019.113692.
[3] X. Wang, P. Chen, and C. Cheng, “Stability and convergency exploration of matrix exponential integration on power delivery network transient simulation,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp. 1–1, 2019. https://doi.org/10.1109/TCAD.2019.2954473.
[4] M. Hochbruck and A. Ostermann, “Exponential integrators,” Acta Numer., vol. 19, pp. 209–286, 2010.
[5] F. R. Gantmacher, The Theory of Matrices. Vol. 1. AMS Chelsea Publishing, Providence, RI, 1998. Translated from the Russian by K. A. Hirsch, Reprint of the 1959 translation.
[6] C. B. Moler and C. F. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,” SIAM Rev., vol. 45, no. 1, pp. 3–49, 2003.
[7] N. J. Higham, Functions of Matrices: Theory and Computation. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.
[8] H. Tal-Ezer, “Spectral methods in time for parabolic problems,” SIAM J. Numer. Anal., vol. 26, no. 1, pp. 1–11, 1989.
[9] H. De Raedt, K. Michielsen, J. S. Kole, and M. T. Figge, “One-step finite-difference time-domain algorithm to solve the Maxwell equations,” Phys. Rev. E, vol. 67, p. 056706, 2003.
[10] T. Schmelzer and L. N. Trefethen, “Evaluating matrix functions for exponential integrators via Carathéodory-Fejér approximation and contour integrals,” Electron. Trans. Numer. Anal., vol. 29, pp. 1–18, 2007/08.
[11] M. Caliari and A. Ostermann, “Implementation of exponential Rosenbrock-type integrators,” *Appl. Numer. Math.*, vol. 59, no. 3-4, pp. 568–581, 2009.

[12] 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.*, vol. 33, no. 2, pp. 488–511, 2011. [http://dx.doi.org/10.1137/100788860]

[13] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, 2d ed., 2003. Available from [http://www-users.cs.umn.edu/~saad/books.html](http://www-users.cs.umn.edu/~saad/books.html)

[14] H. A. van der Vorst, *Iterative Krylov methods for large linear systems*. Cambridge University Press, 2003.

[15] H. A. van der Vorst, “An iterative Krylov method for solving $f(A)x = b$, using Krylov subspace information obtained for the symmetric positive definite matrix $A$,” *J. Comput. Appl. Math.*, vol. 18, pp. 249–263, 1987.

[16] V. L. Druskin and L. A. Knizhnerman, “Two polynomial methods of calculating functions of symmetric matrices,” *U.S.S.R. Comput. Math. Phys.*, vol. 29, no. 6, pp. 112–121, 1989.

[17] L. A. Knizhnerman, “Calculation of functions of unsymmetric matrices using Arnoldi’s method,” *U.S.S.R. Comput. Math. Phys.*, vol. 31, no. 1, pp. 1–9, 1991.

[18] E. Gallopoulos and Y. Saad, “Efficient solution of parabolic equations by Krylov approximation methods,” *SIAM J. Sci. Statist. Comput.*, vol. 13, no. 5, pp. 1236–1264, 1992.

[19] Y. Saad, “Analysis of some Krylov subspace approximations to the matrix exponential operator,” *SIAM J. Numer. Anal.*, vol. 29, no. 1, pp. 209–228, 1992.

[20] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[21] V. L. Druskin, “Krylov subspace exponential time domain solution of Maxwell’s equations in photonic crystal modeling,” *J. Comput. Appl. Math.*, vol. 293, pp. 24–30, 2016. [http://dx.doi.org/10.1016/j.cam.2015.04.022]

[22] H. Zhuang, X. Wang, Q. Chen, P. Chen, and C. Cheng, “From circuit theory, simulation to SPICE-Diego: A matrix exponential approach for time-domain analysis of large-scale circuits,” *IEEE Circuits and Systems Magazine*, vol. 16, no. 2, pp. 16–34, 2016.

[23] W. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer Verlag, 2003.

[24] M. A. Botchev, L. A. Knizhnerman, and E. E. Tyrtyshnikov, “A residual concept for Krylov subspace eigenproblem solution,” *SIAM J. Sci. Comput.*, vol. 23, no. 2, pp. 488–511, 2001.

[25] Y. Saad and M. H. Schultz, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM J. Sci. Statist. Comput.*, vol. 7, no. 3, pp. 856–869, 1986.

[26] V. L. Druskin and L. A. Knizhnerman, “Two polynomial methods of calculating functions of symmetric matrices,” *U.S.S.R. Comput. Math. Phys.*, vol. 29, no. 6, pp. 112–121, 1989.

[27] L. A. Knizhnerman, “Calculation of functions of unsymmetric matrices using Arnoldi’s method,” *U.S.S.R. Comput. Math. Phys.*, vol. 31, no. 1, pp. 1–9, 1991.

[28] E. Gallopoulos and Y. Saad, “Efficient solution of parabolic equations by Krylov approximation methods,” *SIAM J. Sci. Statist. Comput.*, vol. 13, no. 5, pp. 1236–1264, 1992.

[29] Y. Saad, “Analysis of some Krylov subspace approximations to the matrix exponential operator,” *SIAM J. Numer. Anal.*, vol. 29, no. 1, pp. 209–228, 1992.

[30] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[31] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[32] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[33] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[34] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[35] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[36] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[37] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[38] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[39] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.

[40] V. L. Druskin and L. A. Knizhnerman, “Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic,” *Numer. Lin. Alg. Appl.*, vol. 2, pp. 205–217, 1995.
[37] T. A. Davis, “A column pre-ordering strategy for the unsymmetric-pattern multifrontal method,” ACM Trans. Math. Software, vol. 30, no. 2, pp. 167–195, 2004.

[38] Y. Notay and AGMG team, “AGMG: Iterative solution with AGgregation-based algebraic MultiGrid.” Software package, http://agmg.eu.

[39] Y. Notay, “An aggregation-based algebraic multigrid method,” Electronic Transactions on Numerical Analysis, vol. 37, pp. 123–146, 2010.

[40] A. Napov and Y. Notay, “An algebraic multigrid method with guaranteed convergence rate,” SIAM J. Sci. Comput., vol. 32, pp. A1079–A1109, 2012.

[41] Y. Notay, “Aggregation-based algebraic multigrid for convection-diffusion equations,” SIAM J. Sci. Comput., vol. 34, pp. A2288–A2316, 2012.

[42] V. T. Zhukov, “Explicit methods of numerical integration for parabolic equations,” Mathematical Models and Computer Simulations, vol. 3, no. 3, pp. 311–332, 2011.

[43] T. Göckler and V. Grimm, “Uniform approximation of $\phi$-functions in exponential integrators by a rational Krylov subspace method with simple poles,” SIAM Journal on Matrix Analysis and Applications, vol. 35, no. 4, pp. 1467–1489, 2014. http://dx.doi.org/10.1137/140964655.