Electronic Transactions on Numerical Analysis.
Volume 65, pp. 414–440, 2026.

ISSN 1068–9613.
DOI: 10.1553/etna_vol65s414

ETNA
Kent State University and
Johann Radon Institute (RICAM)

A GENERAL FRAMEWORK FOR KRYLOV ODE RESIDUALS WITH APPLICATIONS TO RANDOMIZED KRYLOV METHODS*

EMIL KRIEGER† AND MARCEL SCHWEITZER†

Abstract. Randomized Krylov subspace methods that employ the sketch-and-solve paradigm to substantially reduce orthogonalization costs have recently shown great promise in speeding up computations for many core linear algebra tasks (e.g., solving linear systems, eigenvalue problems and matrix equations, as well as approximating the action of matrix functions on vectors) whenever a nonsymmetric matrix is involved. An important application that requires approximating the action of matrix functions on vectors is the implementation of exponential integration schemes for ordinary differential equations (ODEs). In this paper we specifically analyze randomized Krylov methods from this point of view. In particular, we use the residual of the underlying differential equation to derive a new, reliable a posteriori error estimate that can be used to monitor convergence and to decide when to stop the iteration. To do so, we first develop a very general framework for Krylov ODE residuals that unifies existing results, simplifies their derivation, and allows us to extend the concept to a wide variety of methods beyond the randomized Arnoldi scheme (e.g., rational Krylov methods, Krylov methods using a non-standard inner product, etc.). In addition, we discuss certain aspects regarding the efficient implementation of sketched Krylov methods. Numerical experiments for large-scale ODE models from real-world applications illustrate the use of the sketched residual norm as a stopping criterion as well as the general competitiveness of sketched Krylov methods for ODEs in comparison to other Krylov-based methods.

Key words. Krylov subspace methods, ordinary differential equations, residual, randomized sketching, exponential integration

AMS subject classifications. 65F60, 68W20, 65L05, 65F25

1. Introduction.

Exponential time integration schemes [35] are among the most promising classes of numerical methods for the solution of stiff, large-scale systems of ordinary differential equations (ODEs). Their implementation crucially relies on the ability to efficiently evaluate the action of matrix functions [32] on vectors—the action of the matrix exponential and φ-functions in the case of first-order problems [35] and compositions of square roots, inverses, and trigonometric functions in the case of second-order problems [23, 34]. The growing success of and interest in exponential integrators can therefore, in part, also be attributed to algorithmic advances for matrix functions in the last two decades [1, 19, 21, 28, 33].

When $A \in \mathbb{R}^{n\times n}$ is large and sparse (or otherwise structured), the matrix function $f(A)$ cannot be computed explicitly due to the enormous computational cost and memory requirements. Instead, one aims to directly approximate $f(A)\boldsymbol{b}$, where $\boldsymbol{b} \in \mathbb{R}^n$ is a given vector, by an iterative method—typically a Krylov subspace method. When A is of moderate size or highly structured such that shifted linear systems $(A + \sigma I)\boldsymbol{x} = \boldsymbol{b}$ can be efficiently treated by a direct solver, rational Krylov subspace methods are often the method of choice [25, 27]. In other cases, e.g., when A is given only implicitly by some routine that returns the result of the matrix-vector product $\boldsymbol{v} \mapsto A\boldsymbol{v}$, one is restricted to the use of polynomial Krylov methods [18, 22, 44, 46], which typically converge much more slowly. This can be a huge problem—in particular when A is nonsymmetric—as storage requirements and orthogonalization costs grow with the number of iterations.

One possible remedy for this problem is the use of an *incomplete orthogonalization* within Krylov methods (instead of the usual, full Gram–Schmidt orthogonalization that is typically

*Received March 12, 2026. Accepted July, 1 2026. Published online on August 18, 2026. Recommended by Stefan Güttel. This work was supported by Deutsche Forschungsgemeinschaft through DFG grant 538686094 – "Matrix functions via randomized sketching".

†Corresponding author: M. Schweitzer. School of Mathematics and Natural Sciences, Bergische Universität Wuppertal, 42097 Wuppertal, Germany ({krieger,marcel}@uni-wuppertal.de).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

used in the Arnoldi method), which has been demonstrated to work quite successfully in the context of exponential integrators [22, 37]. It does, however, typically lead to a further delay in convergence and can necessitate the use of restarting (or "sub-time stepping") techniques in order to reach the desired accuracy.

A relatively recent new avenue in algorithms for approximating the action of a matrix function on a vector is the use of *randomized methods*, in particular those based on the *sketch-and-solve paradigm* [39, 48]. In the context of Krylov methods, these approaches typically aim at substantially reducing orthogonalization costs while avoiding the delay of convergence observed in methods with incomplete orthogonalization; see, e.g., [4, 12, 15, 24, 29, 30, 40, 42, 43] and the references therein.

While these methods have shown very promising results experimentally, they are still in their infancy. There is still a large gap in comparison to classical Krylov methods without randomization, both regarding their theoretical understanding and their adoption in production-level codes. One particular shortcoming that might prevent more widespread adoption is the lack of reliable error estimates and stopping criteria. Most publications on this subject so far either completely omit a discussion of this topic or employ a heuristic error estimate based on the difference of iterates [24, 29]. Such an estimate can already be unreliable for classical Krylov methods and even more so in the presence of randomization.

For matrix functions that are connected to an underlying (first- or second-order) ODE, an established way to monitor the progress of the method is to keep track of the *ODE residual*. For classical Krylov methods, it is well known that the residual (or its norm) can be cheaply computed in terms of quantities arising naturally from the Arnoldi process [8, 9, 10, 11, 13]. In this paper we introduce a general framework for Krylov residuals, which in particular allows us to adapt the above-mentioned results in such a way that they can straightforwardly be transferred to randomized Krylov methods. We further show that the Krylov approximation error norm can be rigorously bounded in terms of the sketched residual norm, demonstrating that the norm of the sketched residual can be used as a reliable, non-heuristic stopping criterion.

In addition to the contributions mentioned above, we discuss some possible refinements in the implementation of sketched Krylov methods that can improve their efficiency and numerical stability.

The remainder of this paper is organized as follows. In Section 2, we introduce basic material on Krylov subspace methods for matrix functions, before specifically focusing on problems with an underlying ODE. Based on this, we introduce a general framework for Krylov-based ODE residuals in Section 3, which unifies existing results from the literature and allows us to obtain conceptually simpler proofs for important properties. We specifically apply this framework to sketched Krylov methods in Section 4 and illustrate how it can be used to obtain a rigorous a posteriori error estimate for these methods. In this section, we also discuss the efficient implementation of residual-based sketched Krylov methods, focusing in particular on the possibility of introducing restarts in order to improve numerical stability and efficiency. Numerical experiments for several large-scale real-world problems are presented in Section 5. Concluding remarks and an outlook on potential future research topics are given in Section 6.

1.1. Notation and conventions. Throughout the paper we use the following notation: By $\langle \cdot, \cdot \rangle$, we denote the Euclidean inner product, and by $\| \cdot \|$ we denote the vector or matrix norm it induces. The transpose of a matrix M is denoted by $M^\top$. The spectrum of a matrix M is denoted by $\sigma(M)$. The imaginary part of a complex number z is denoted by $\Im(z)$. The mth canonical unit vector is denoted by $\boldsymbol{e}_m$. The ith entry of a vector $\boldsymbol{v}$ is denoted by $[\boldsymbol{v}]_i$, and when indexing vectors or matrices, we sometimes use "MATLAB-style notation", i.e., $i : j$ refers to all indices $i, i+1, \ldots, j-1, j$. In the context of block Krylov methods (with block

ETNA
Kent State University and
Johann Radon Institute (RICAM)

size ℓ), we denote "block vectors" $\mathcal{U} \in \mathbb{R}^{n\times\ell}$ by uppercase calligraphic letters and "block scalars" $\mathcal{g} \in \mathbb{R}^{\ell\times\ell}$ by lowercase calligraphic letters.

For ease of exposition we restrict ourselves to real-valued matrices and vectors throughout the manuscript, although most results are rather straightforwardly transferable to the complex-valued case.

2. Basic material. In this section we present basic material on matrix functions and Krylov subspace methods, as well as their application in the context of solving ODEs.

2.1. Functions of matrices. Matrix functions can be defined in many different ways. The most common definitions are based on the Jordan canonical form, Hermite interpolating polynomials, or the Cauchy integral formula; we refer to [32, Section 1.2] for a thorough treatment.

As we are mostly interested in entire functions in this work, we only recall the definition based on the Cauchy integral formula, which applies to functions f that are analytic in a region $\Omega \subseteq \mathbb{C}$ containing the spectrum $\sigma(A)$ of $A \in \mathbb{R}^{n\times n}$. We can then define $f(A)$ via

$$f(A) := \frac{1}{2\pi i}\int_\Gamma f(z)(zI - A)^{-1}\,\mathrm{d}z,$$

where $\Gamma \subset \Omega$ is a path that winds around the spectrum of A exactly once.

It is important to note that even when A is very sparse, $f(A)$ is, in general, a dense matrix, so that for large n it is not feasible to explicitly form $f(A)$ because it is often impossible to store this matrix (in addition to being very costly to compute). Therefore, in applications where the action of $f(A)$ on a vector $\boldsymbol{b}$ is required—such as in exponential integrators—one typically aims to directly approximate $f(A)\boldsymbol{b}$ by an iterative method. The most popular such methods are *Krylov subspace methods*, which we briefly introduce next.

2.2. Krylov methods for matrix functions. The backbone of Krylov methods for matrix functions is the *Arnoldi method* [3], which computes an orthonormal basis (ONB) of the Krylov subspace $\mathcal{K}_m(A,\boldsymbol{b}) := \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, \dots, A^{m-1}\boldsymbol{b}\}$ via a modified Gram–Schmidt approach.

After m iterations, the Arnoldi method yields a matrix $U_m = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_m] \in \mathbb{R}^{n\times m}$ that contains a nested orthonormal basis of $\mathcal{K}_m(A,\boldsymbol{b})$ and an upper Hessenberg matrix $\underline{G}_m \in \mathbb{R}^{m+1\times m}$ containing the orthogonalization coefficients. These matrices are connected by the *Arnoldi relation*

$$AU_m = U_{m+1}\underline{G}_m = U_m G_m + g_{m+1,m}\boldsymbol{u}_{m+1}\boldsymbol{e}_m^\top, \tag{2.1}$$

where $G_m \in \mathbb{R}^{m\times m}$ is the upper $m \times m$ part of $\underline{G}_m$, i.e.,

$$\underline{G}_m = \begin{bmatrix} G_m \\ g_{m+1,m}\boldsymbol{e}_m^\top \end{bmatrix}.$$

Clearly, (2.1) directly implies

$$U_m^\top A U_m = G_m$$

due to the orthogonality of the columns of U_m. Based on these quantities, for a general matrix function $f(A)\boldsymbol{b}$, the *Arnoldi (or FOM) approximation* is defined as

$$f(A)\boldsymbol{b} \approx \boldsymbol{f}_m = U_m f\big(U_m^\top A U_m\big)U_m^\top \boldsymbol{b} = U_m f(G_m)\boldsymbol{e}_1\|\boldsymbol{b}\|. \tag{2.2}$$

With $\operatorname{nnz}(A)$ denoting the number of nonzeros in A, the Arnoldi method generally has a computational cost of $\mathcal{O}\big(\operatorname{nnz}(A)m + nm^2\big)$ operations, where the first term accounts for

ETNA
Kent State University and
Johann Radon Institute (RICAM)

the matrix-vector products, while the second term accounts for the modified Gram–Schmidt orthogonalization. In many practical situations, e.g., when A results from the finite difference or finite element discretization of a differential operator, nnz(A) $= \mathcal{O}(n)$, so that the orthogonalization causes the dominant computational cost. In addition, storing the (dense) ONB requires $\mathcal{O}(nm)$ memory, which can also be prohibitive if a large number m of iterations is required to reach the desired accuracy.

There exist different theoretical justifications for why (2.2) is a sensible approximation of $f(A)\boldsymbol{b}$. We recapitulate two of the most important and widely used ones here. First, the Arnoldi iteration can alternatively be characterized as

$$\boldsymbol{f}_m = p_{m-1}(A)\boldsymbol{b},$$

where p_{m-1} is the polynomial that interpolates f at the spectrum of G_m (i.e., at the so-called Ritz values) in the Hermite sense; see [44, Theorem 3.3]. Second, the Arnoldi approximation satisfies

$$\|f(A)\boldsymbol{b} - \boldsymbol{f}_m\| \leq 2(1+\sqrt{2}) \min_{p \in \Pi_{m-1}} \max_{z \in W(A)} \|f(z) - p(z)\|, \tag{2.3}$$

where Π_{m-1} denotes the set of all polynomials of degree at most $m-1$ and $W(A)$ is the numerical range (or field of values) of A. Inequality (2.3) easily follows from the Crouzeix–Palencia theorem [16] and allows us to relate the convergence of the Arnoldi approximation of $f(A)\boldsymbol{b}$ to the convergence of polynomial approximations of the scalar function f on $W(A)$; see, e.g., [6].

In the context of solving ODEs, an alternative way of deriving and justifying the Arnoldi approximation (2.2) arises from imposing a Galerkin condition, as we will illustrate next.

2.3. Krylov methods for ODEs. For illustration purposes, in this section we recapitulate the Krylov solution of linear, homogeneous, first-order initial value problems (IVPs)

$$\begin{cases} \boldsymbol{y}'(t) = -A\boldsymbol{y}(t) & \text{on } (0,T], \\ \boldsymbol{y}(0) = \boldsymbol{b}_0, \end{cases} \tag{2.4}$$

where $A \in \mathbb{R}^{n \times n}$, as this is the simplest model problem fitting into the framework developed in Section 3. It is well known that the solution of (2.4) can be written in terms of the matrix exponential as

$$\boldsymbol{y}(t) = \exp(-tA)\boldsymbol{b}_0 \qquad \text{for all } t \in [0,T].$$

Assume we want to find an approximate solution $\boldsymbol{y}_m$ of (2.4) in the Krylov subspace $\mathcal{K}_m(A, \boldsymbol{b}_0)$, i.e., $\boldsymbol{y}_m : [0,T] \to \mathcal{K}_m(A, \boldsymbol{b}_0)$. We introduce the corresponding ODE residual

$$\boldsymbol{r}_m(t) := -\boldsymbol{y}_m'(t) - A\boldsymbol{y}_m(t) \qquad t \in [0,T].$$

Clearly, $\boldsymbol{r}_m(t) \equiv \boldsymbol{0}$ would imply that $\boldsymbol{y}_m$ solves the IVP (2.4).

One possible approach for defining an approximate solution $\boldsymbol{y}_m(t)$ is to impose a Galerkin condition, i.e., one demands that

$$\boldsymbol{r}_m(t) \perp \mathcal{K}_m(A, \boldsymbol{b}_0) \qquad \text{for all } t \in [0,T]. \tag{2.5}$$

Using the notation from Section 2.2, in particular letting U_m be an ONB of $\mathcal{K}_m(A, \boldsymbol{b}_0)$, we can write $\boldsymbol{y}_m(t) = U_m \boldsymbol{x}_m(t)$ for some $\boldsymbol{x}_m(t) : [0,T] \to \mathbb{R}^m$. Condition (2.5) then implies

$$\boldsymbol{0} = U_m^\top \boldsymbol{r}_m(t) = -U_m^\top (U_m \boldsymbol{x}_m'(t) - AU_m \boldsymbol{x}_m(t)) = -\boldsymbol{x}_m'(t) - G_m \boldsymbol{x}_m(t),$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

i.e., $\boldsymbol{x}_m$ solves the projected IVP

$$\begin{cases} \boldsymbol{x}_m'(t) = -G_m \boldsymbol{x}_m(t) & \text{on } (0,T], \\ \boldsymbol{x}_m(0) = \boldsymbol{e}_1 \|\boldsymbol{b}_0\|, \end{cases} \tag{2.6}$$

where the initial conditions are established by requiring that $\boldsymbol{y}_m(0) = \boldsymbol{b}_0$, i.e., that the approximation fulfills the initial conditions of the original IVP (2.4). The solution of (2.6) is given by

$$\boldsymbol{x}_m(t) = \exp(-tG_m)\boldsymbol{e}_1 \|\boldsymbol{b}_0\|,$$

so that $\boldsymbol{y}_m(t) = U_m \exp(-tG_m)\boldsymbol{e}_1 \|\boldsymbol{b}_0\|$ is exactly the Arnoldi approximation (2.2) of $f(z) = \exp(-tz)$, providing another possible justification for why this is a sensible approximation. A very attractive feature of the Arnoldi approximation in the context of ODEs is that the norm of its residual can be used as a reliable and easy-to-compute stopping criterion. It is straightforward to verify that

$$\|\boldsymbol{r}_m(t)\| = |g_{m+1,m}[\boldsymbol{x}_m(t)]_m| \tag{2.7}$$

(see, e.g., [8, 9]), which can be readily computed from quantities available from the Arnoldi process. That the residual norm is indeed a suitable stopping criterion for the iteration can be seen as follows: The ODE residual and the *error*

$$\boldsymbol{\xi}_m(t) := \boldsymbol{y}(t) - \boldsymbol{y}_m(t)$$

are related by the semilinear IVP

$$\begin{cases} \boldsymbol{\xi}_m'(t) = -A\boldsymbol{\xi}_m(t) + \boldsymbol{r}_m(t) & \text{on } (0,T], \\ \boldsymbol{\xi}_m(0) = \boldsymbol{0}, \end{cases}$$

so that, using the variation-of-constants formula, the error can be written as

$$\boldsymbol{\xi}_m(t) = \int_0^t \exp(-(t-\theta)A)\boldsymbol{r}_m(s)\,\mathrm{d}\theta. \tag{2.8}$$

Under the standard assumption that there exist constants $C_1 > 0$ and $\omega_1 \in \mathbb{R}$ such that

$$\|\exp(-tA)\| \le C_1 e^{-t\omega_1} \qquad \text{for all } t \ge 0, \tag{2.9}$$

one directly obtains the bound

$$\|\boldsymbol{\xi}_m(t)\| \le C_1 t \varphi_1(-t\omega_1) \max_{0\le\theta\le t} \|\boldsymbol{r}_m(\theta)\|, \qquad \text{where} \quad \varphi_1(z) = \frac{e^z - 1}{z}, \tag{2.10}$$

from (2.8); see [8, Lemma 4.1]. Thus, a *uniform* bound for the residual norm on $[0,T]$ directly implies a bound for the *error norm*.

Note that, e.g., when $-A$ is a stable operator with field of values in the left half-plane, condition (2.9) is satisfied with $\omega_1 = 0$ and $C_1 = 1$. More generally, ω_1 can be taken as the left endpoint of the field of values of A.

Due to the features mentioned above, it is very attractive to work with residual-based methodologies when solving large-scale ODEs using Krylov subspace methods, and there is a large body of work on this topic [7, 8, 9, 10, 11, 13]. In the next section, we aim to unify and generalize existing results in a framework that allows us to extend it to other, more general situations and in particular to Krylov methods based on randomized sketching (cf. Section 4).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

3. A general framework for Krylov ODE residuals. In this section we consider a general semilinear differential equation of the form

$$\boldsymbol{y}^{(p)}(t) = -A\boldsymbol{y}(t) + \boldsymbol{w}(t) \qquad \text{for } t \in (0, T], \tag{3.1}$$

where $p \in \mathbb{N}$ and $\boldsymbol{w}$ is a vector-valued function $\boldsymbol{w}(t) : (0, T] \to \mathbb{R}^n$. We assume that we are given the initial conditions

$$\boldsymbol{y}(0) = \boldsymbol{b}_0, \quad \boldsymbol{y}'(0) = \boldsymbol{b}_1, \quad \ldots, \quad \boldsymbol{y}^{(p-1)}(0) = \boldsymbol{b}_{p-1},$$

which, together with (3.1), define an initial value problem.

3.1. Efficiently computing the Krylov ODE residual and its norm. There has been much work on obtaining computationally feasible formulas for the residual of Krylov solutions for (3.1), for many different special cases. For example, the papers [8, 9] treat the case $p = 1$ and $\boldsymbol{w}(t) \equiv \mathbf{0}$, the papers [11, 13] treat $p = 1$ and a constant inhomogeneity $\boldsymbol{w}(t) \equiv \boldsymbol{w} \neq \mathbf{0}$, whereas [7, 10] consider $p = 2$. In each case, formulas for the residual are derived by explicit, lengthy computations that need to be adapted and redone from scratch whenever the structure of the ODE or the employed method changes.

In the following we present a general framework that exposes the principles underlying all approaches mentioned above and therefore allows us to extend them to other differential equations (and, in particular, to other algorithms) with minimal effort. An additional benefit is that the proofs of the results simplify in comparison to those presented in some of the references above.

We assume that we have the subspaces

$$\mathcal{V}_m \subseteq \mathcal{V}_{m+1} \subseteq \mathbb{R}^n$$

of dimension $\dim(\mathcal{V}_i) = d_i$ and that we have the inclusion $A\mathcal{V}_m \subseteq \mathcal{V}_{m+1}$. We further assume that $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_{p-1} \in \mathcal{V}_m$ as well as $\boldsymbol{w}(t) \in \mathcal{V}_m$ for all $t \in (0, T]$.[1]

Let $\boldsymbol{y}_m : [0, T] \to \mathcal{V}_m$ be an approximate solution of (3.1) determined by the Galerkin condition

$$\boldsymbol{r}_m(t) \perp_* \mathcal{V}_m \qquad \text{for all } t \in [0, T], \tag{3.2}$$

with respect to some general inner product $\langle \cdot, \cdot \rangle_*$ on $\mathcal{V}_{m+1}$, where $\boldsymbol{r}_m$ is the residual

$$\boldsymbol{r}_m(t) = -\boldsymbol{y}_m^{(p)}(t) - A\boldsymbol{y}_m(t) + \boldsymbol{w}(t).$$

In the following, we denote by U_m an orthonormal basis of $\mathcal{V}_m$ with respect to $\langle \cdot, \cdot \rangle_*$ and assume that $\mathcal{U}_{m+1} \in \mathbb{R}^{n \times (d_{m+1} - d_m)}$ is such that $U_{m+1} = [U_m, \mathcal{U}_{m+1}]$ is an orthonormal basis of $\mathcal{V}_{m+1}$.[2] The adjoint of a matrix M with respect to $\langle \cdot, \cdot \rangle_*$ is denoted by M^* in the following. Using this notation, we can write

$$\boldsymbol{y}_m(t) = U_m \boldsymbol{x}_m(t) \qquad \text{with some } \boldsymbol{x}_m : [0, T] \to \mathbb{R}^m,$$

and the corresponding residual is thus given by

$$\boldsymbol{r}_m(t) = -U_m \boldsymbol{x}_m^{(p)}(t) - AU_m \boldsymbol{x}_m(t) + \boldsymbol{w}(t). \tag{3.3}$$

[1] The assumption that $\mathcal{V}_m$ must include all initial conditions as well as $\boldsymbol{w}(t)$ at all time points seems quite restrictive at first. However, for most practically relevant cases, only a few vectors are required to actually lie in $\mathcal{V}_{m-1}$. This is discussed later in Section 3.2.

[2] In the case $d_i = \dim(\mathcal{V}_i) = i$, as, for example, for standard Krylov subspaces, we have $d_{m+1} - d_m = 1$. We allow for more general dimensions of the nested subspaces such that block Krylov approaches and other, more general methods, are also captured by the framework.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Any method fitting into the above framework can be interpreted as being based on solving a certain projected version of the initial value problem under consideration, as we outline in the following. Left-multiplying (3.3) by U_m^* and applying the Galerkin condition (3.2) yields

$$\mathbf{0} = U_m^* \boldsymbol{r}_m(t) = -\boldsymbol{x}_m^{(p)}(t) - U_m^* A U_m \boldsymbol{x}_m(t) + U_m^* \boldsymbol{w}(t).$$

Demanding that the approximate solution $\boldsymbol{y}_m$ exactly satisfies the initial conditions posed at $t = 0$, i.e., $\boldsymbol{y}_m^{(j)}(0) = \boldsymbol{b}_j, j = 0, \ldots, p-1$, further yields

$$U_m \boldsymbol{x}_m^{(j)}(0) = \boldsymbol{b}_j, \qquad j = 0, \ldots, p-1.$$

Again left-multiplying by U_m^* gives the projected initial conditions $\boldsymbol{x}_m^{(j)}(0) = U_m^* \boldsymbol{b}_j$. In summary, we obtain the projected IVP

$$\begin{cases} \boldsymbol{x}_m^{(p)}(t) = -U_m^* A U_m \boldsymbol{x}_m(t) + U_m^* \boldsymbol{w}(t) & \text{on } (0, T], \\ \boldsymbol{x}_m^{(j)}(0) = U_m^* \boldsymbol{b}_j, \qquad j = 0, \ldots, p-1, \end{cases} \tag{3.4}$$

that determines the function $\boldsymbol{x}_m(t)$. Note that in the case of the standard Arnoldi method applied to a homogeneous first-order ODE, the IVP (3.4) agrees with (2.6).

THEOREM 3.1. *Let*

(i) $\mathcal{V}_m \subseteq \mathcal{V}_{m+1}$ *with nested orthonormal bases* U_m, U_{m+1} *such that* $A\mathcal{V}_m \subseteq \mathcal{V}_{m+1}$,
(ii) $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_{p-1} \in \mathcal{V}_m$ *as well as* $\boldsymbol{w}(t) \in \mathcal{V}_m$ *for all* $t \in (0, T]$,
(iii) $\boldsymbol{y}_m(t) := U_m \boldsymbol{x}_m(t)$ *be an approximate solution of* (3.1) *determined by the projected IVP* (3.4) *obtained via the Galerkin condition* (3.2).

Then,

$$\boldsymbol{r}_m(t) = \mathcal{U}_{m+1} \boldsymbol{\beta}_m(t) \qquad \text{with} \quad \boldsymbol{\beta}_m(t) = -\mathcal{U}_{m+1}^* A U_m \boldsymbol{x}_m(t).$$

If, furthermore, there is a subspace $\mathcal{V}_{m-1} \subseteq \mathcal{V}_m$ *with basis* U_{m-1} *such that* $A\mathcal{V}_{m-1} \subseteq \mathcal{V}_m$ *and* $U_m = [U_{m-1}, \mathcal{U}_m]$, *then the expression for* $\boldsymbol{\beta}_m(t)$ *simplifies to*

$$\boldsymbol{\beta}_m(t) = -\mathcal{U}_{m+1}^* A \mathcal{U}_m [\boldsymbol{x}_m(t)]_{d_{m-1}+1:d_m},$$

where $\dim(\mathcal{V}_i) = d_i, i \in \{m-1, m\}$.

Proof. Due to the assumptions that $A\mathcal{V}_m \subseteq \mathcal{V}_{m+1}$ and $\boldsymbol{w}(t) \in \mathcal{V}_m \subseteq \mathcal{V}_{m+1}$, we clearly have that $\boldsymbol{r}_m(t) \in \mathcal{V}_{m+1}$, so that $\boldsymbol{r}_m(t) = U_{m+1} \boldsymbol{c}_{m+1}(t)$ with some time-dependent coefficient vector $\boldsymbol{c}_{m+1}(t)$. The Galerkin condition (3.2) implies

$$\begin{aligned} \boldsymbol{c}_{m+1}(t) = U_{m+1}^* \boldsymbol{r}_m(t) &= \begin{bmatrix} U_m^* \boldsymbol{r}_m(t) \\ \mathcal{U}_{m+1}^* \boldsymbol{r}_m(t) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} \\ \mathcal{U}_{m+1}^* (-U_m \boldsymbol{x}_m^{(p)}(t) - A U_m \boldsymbol{x}_m(t) + \boldsymbol{w}(t)) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} \\ -\mathcal{U}_{m+1}^* A U_m \boldsymbol{x}_m(t) \end{bmatrix}, \end{aligned}$$

where we have used the fact that the columns of $\mathcal{U}_{m+1}$ are orthogonal to U_m and $\boldsymbol{w}(t)$ in the last equality. This proves the first assertion.

For the second assertion, note that under the additional assumptions, the columns of AU_{m-1} lie in $\mathcal{V}_m$ and are therefore orthogonal to the columns of $\mathcal{U}_{m+1}$. This immediately gives

$$\mathcal{U}_{m+1}^* A U_m = [\mathcal{U}_{m+1}^* A U_{m-1}, \mathcal{U}_{m+1}^* A \mathcal{U}_m] = [\mathbf{0}^*, \mathcal{U}_{m+1}^* A \mathcal{U}_m],$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

i.e., $-\mathcal{U}_{m+1}^* A U_m \boldsymbol{x}_m(t) = -\mathcal{U}_{m+1}^* A \mathcal{U}_m [\boldsymbol{x}_m(t)]_{d_{m-1}+1:d_m}$. ∎

Denoting by $\|\cdot\|_*$ the norm induced by $\langle\cdot,\cdot\rangle_*$, Theorem 3.1 allows us to easily obtain $\|\boldsymbol{r}_m(t)\|_*$.

COROLLARY 3.2. *Using the notation of Theorem 3.1, we have*

$$\|\boldsymbol{r}_m(t)\|_* = \|\boldsymbol{\beta}_m(t)\|.$$

Proof. Using the result of Theorem 3.1, the assertion follows from the straightforward computation

$$\begin{aligned}\|\boldsymbol{r}_m(t)\|_*^2 &= \langle \boldsymbol{r}_m(t), \boldsymbol{r}_m(t)\rangle_* = \langle \mathcal{U}_{m+1}\boldsymbol{\beta}_m(t), \mathcal{U}_{m+1}\boldsymbol{\beta}_m(t)\rangle_* \\ &= \langle \boldsymbol{\beta}_m(t), \boldsymbol{\beta}_m(t)\rangle = \|\boldsymbol{\beta}_m(t)\|^2. \end{aligned}$$ ∎

We now first discuss the question which inhomogeneities $\boldsymbol{w}(t)$ are naturally covered by our framework before providing several examples of its application in Section 3.3.

3.2. Non-constant inhomogeneities. For arbitrary vector-valued inhomogeneities $\boldsymbol{w}(t)$, the requirement that $\boldsymbol{w}(t) \in \mathcal{V}_m$ for all t in Theorem 3.1 might be very restrictive and may not be satisfied by typical approximation spaces. We are, however, primarily interested in situations in which the solution of the IVP can be expressed in terms of certain matrix functions in order to then apply (rational) Krylov methods. In these situations we typically have, for a moderate value of q,

$$\boldsymbol{w}(t) = \sum_{i=1}^{q} g_i(t)\boldsymbol{w}_i, \tag{3.5}$$

with $g_i : (0,T] \to \mathbb{R}$, $i = 1,\dots,q$, i.e., $\boldsymbol{w}(t)$ is a linear combination (with time-dependent coefficients) of only a few constant vectors. In this case, the solution to the IVP can often be written in the form

$$\boldsymbol{y}(t) = \sum_{i=0}^{p-1} f_i^{\mathrm{IC}}(t,A)\boldsymbol{b}_i + \sum_{j=1}^{q} f_j^{\mathrm{S}}(t,A)\boldsymbol{w}_j, \tag{3.6}$$

with time-dependent matrix functions

$$f_i^{\mathrm{IC}}, f_j^{\mathrm{S}} : [0,T] \times \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}, \qquad i = 0,\dots,p-1, \quad j = 1,\dots,q.$$

The arguably most important special case of inhomogeneities of the form (3.5) arises in initial value problems defining the so-called φ-functions

$$\varphi_j(z) = \sum_{i=0}^{\infty} \frac{z^i}{(i+j)!}$$

that form the basis of most exponential integration schemes [35, 41]: The IVP

$$\begin{cases} \boldsymbol{y}'(t) = -A\boldsymbol{y}(t) + \displaystyle\sum_{j=1}^{q} \frac{t^{j-1}}{(j-1)!}\boldsymbol{w}_j & \text{on } (0,T], \\ \boldsymbol{y}(0) = \boldsymbol{b}_0, \end{cases} \tag{3.7}$$

is solved by

$$\boldsymbol{y}(t) = \exp(-tA)\boldsymbol{b}_0 + t\varphi_1(-tA)\boldsymbol{w}_1 + t^2\varphi_2(-tA)\boldsymbol{w}_2 + \cdots + t^q\varphi_q(-tA)\boldsymbol{w}_q.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

In principle, (3.6) could, e.g., be approximated using a block Krylov method with starting block $\mathcal{B} = [\boldsymbol{b}_0, \dots, \boldsymbol{b}_{p-1}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_q]$ (cf. Example 3.5). Often, however, it is beneficial to first rewrite the IVP by splitting $\boldsymbol{y}(t) = \widetilde{\boldsymbol{y}}(t) + \boldsymbol{a}(t)$ such that $\widetilde{\boldsymbol{y}}^{(j)} = \boldsymbol{0}, j = 0, \dots, p-1$, leading to

$$\begin{cases} \widetilde{\boldsymbol{y}}^{(p)}(t) = -A\widetilde{\boldsymbol{y}}(t) + \boldsymbol{w}(t) - A\boldsymbol{a}(t) - \boldsymbol{a}^{(p)}(t) & \text{on } (0,T], \\ \widetilde{\boldsymbol{y}}(0) = \widetilde{\boldsymbol{y}}'(0) = \cdots = \widetilde{\boldsymbol{y}}^{(p-1)}(0) = \boldsymbol{0}. \end{cases}$$

For example, for (3.7), we can set $\boldsymbol{a}(t) := \sum_{j=0}^{q-1} \frac{t^j}{j!} \widetilde{\boldsymbol{w}}_j$ with the recursively defined vectors

$$\widetilde{\boldsymbol{w}}_0 = \boldsymbol{b}_0, \qquad \widetilde{\boldsymbol{w}}_j = -A\widetilde{\boldsymbol{w}}_{j-1} + \boldsymbol{w}_j, \quad j = 1, \dots, q$$

(cf. [41]), which leads to the IVP

$$\begin{cases} \widetilde{\boldsymbol{y}}'(t) = -A\widetilde{\boldsymbol{y}}(t) + \dfrac{t^{q-1}}{(q-1)!}\widetilde{\boldsymbol{w}}_q & \text{on } (0,T], \\ \widetilde{\boldsymbol{y}}(0) = \boldsymbol{0}. \end{cases} \tag{3.8}$$

The solution of (3.8) is given by $\widetilde{\boldsymbol{y}}(t) = t^q \varphi_q(-tA)\widetilde{\boldsymbol{w}}_q$ and can thus be approximated in the Krylov space $\mathcal{K}_m(A, \widetilde{\boldsymbol{w}}_q)$.

3.3. Examples. Theorem 3.1 and Corollary 3.2 can be applied under rather general conditions and allow us to reproduce several known results as well as to obtain generalizations to many other important situations, as we illustrate in the following examples.

EXAMPLE 3.3. Consider the standard polynomial Arnoldi method for the first-order IVP

$$\begin{cases} \boldsymbol{y}'(t) = -A\boldsymbol{y}(t) & \text{on } (0,T], \\ \boldsymbol{y}(0) = \boldsymbol{b}_0, \end{cases}$$

which we considered in Section 2.3. Here, $\mathcal{V}_m = \mathcal{K}_m(A, \boldsymbol{b}_0)$, the nested orthonormal basis is $U_{m+1} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_{m+1}]$, and the Euclidean inner product is used, i.e., $\langle \cdot, \cdot \rangle_* = \langle \cdot, \cdot \rangle$. In this case, the projected IVP arising from the Galerkin condition (3.2) is (2.6), and the vector $\boldsymbol{y}_m(t) = U_m \exp(-tG_m)\boldsymbol{e}_1 \|\boldsymbol{b}_0\|$ is exactly the standard Arnoldi approximation of the exponential. Due to $\boldsymbol{u}_{m+1}^* A \boldsymbol{u}_m = g_{m+1,m}$, Theorem 3.1 and Corollary 3.2 directly yield the well-known result (2.7).

EXAMPLE 3.4. Our framework allows us to effortlessly obtain the exact same results as in Example 3.3 also when a non-standard inner product is used. In certain applications, switching to a different inner product allows one to measure the error in a more natural way and can additionally have algorithmic advantages. As an example, consider the Poisson system

$$\begin{cases} \boldsymbol{y}'(t) = JQ\boldsymbol{y}(t) & \text{on } (0,T], \\ \boldsymbol{y}(0) = \boldsymbol{b}_0, \end{cases} \tag{3.9}$$

with quadratic Hamiltonian $\mathcal{H}(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^\top Q \boldsymbol{y}$, where Q is symmetric positive definite and the structure matrix J is skew-symmetric, i.e., $J^\top = -J$. For such a problem, it is natural to measure the error (or residual) norm using the inner product $\langle \boldsymbol{v}, \boldsymbol{w} \rangle_Q = \langle Q\boldsymbol{v}, \boldsymbol{w} \rangle$ induced by Q. In addition, the so-called Q-Arnoldi method (i.e., the Arnoldi method using $\langle \cdot, \cdot \rangle_Q$ instead of the Euclidean inner product) has two further important advantages; see, e.g., [38]. First, the system matrix JQ is skew-self-adjoint with respect to the Q-inner product so that the

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Krylov basis vectors satisfy a three-term recurrence akin to the Lanczos process. Second, the corresponding Krylov approximations

$$\boldsymbol{y}_m(t) := U_m \exp(tG_m)\boldsymbol{e}_1 \|\boldsymbol{b}_0\|_Q \tag{3.10}$$

are energy-preserving, i.e., $\mathcal{H}(\boldsymbol{y}_m(t)) = \mathcal{H}(\boldsymbol{y}_m(0))$ for all $t > 0$, as it is also the case for the exact solution of the Poisson system (3.9). Analogously to Example 3.3, we obtain

$$\|\boldsymbol{r}_m(t)\|_Q = |g_{m+1,m}| \, \|\boldsymbol{b}_0\|_Q \, |[\exp(tG_m)]_{m,1}| \,,$$

where G_m is the tridiagonal matrix arising from the Q-Arnoldi process. Such a relation is certainly of value: In [38, Section 3.3], the authors mention the lack of a natural concept for monitoring the progress of the method as a main reason for not using (3.10) in practice and resorting to an approach based on solving linear systems instead.

EXAMPLE 3.5. The solution of the second-order IVP

$$\begin{cases} \boldsymbol{y}''(t) = -A\boldsymbol{y}(t) + \boldsymbol{w} & \text{on } (0,T], \\ \boldsymbol{y}(0) = \boldsymbol{0}, \quad \boldsymbol{y}'(0) = \boldsymbol{b}_1, \end{cases}$$

is given by

$$\boldsymbol{y}(t) = A^{-1}\left(I - \cos\left(t\sqrt{A}\right)\right)\boldsymbol{w} + \left(\sqrt{A}\right)^{-1} \sin\left(t\sqrt{A}\right)\boldsymbol{b}_1. \tag{3.11}$$

The approach proposed in [7] for approximating (3.11) is to use the block Krylov subspace

$$\mathcal{K}_m^{\square}(A,\mathcal{B}) = \operatorname{colspan}\left(\mathcal{B}, A\mathcal{B}, A^2\mathcal{B}, \ldots, A^{m-1}\mathcal{B}\right)$$

with $\mathcal{B} = [\boldsymbol{w}, \boldsymbol{b}_1]$. Just as for the standard Arnoldi method, we denote the orthonormal basis of $\mathcal{K}_m^{\square}(A,\mathcal{B})$ computed by the block Arnoldi method by $U_m \in \mathbb{R}^{n\times 2m}$ and the compression of A onto the block Krylov space as $G_m = U_m^\top A U_m \in \mathbb{R}^{2m\times 2m}$, and we impose a Galerkin condition $\boldsymbol{r}_m(t) \perp \mathcal{K}_m^{\square}(A,\mathcal{B})$ for the residual $\boldsymbol{r}_m(t) = -\boldsymbol{y}_m''(t) - A\boldsymbol{y}_m(t) + \boldsymbol{w}$. The projected IVP (3.4) resulting from this condition reads

$$\begin{cases} \boldsymbol{x}_m''(t) = -G_m\boldsymbol{x}_m(t) + U_m^\top\boldsymbol{w} & \text{on } (0,T], \\ \boldsymbol{x}_m(0) = \boldsymbol{0}, \quad \boldsymbol{x}_m' = U_m^\top\boldsymbol{b}_1, \end{cases}$$

and is solved by

$$\boldsymbol{x}_m(t) = \left(G_m\right)^{-1}\left(I - \cos\left(t\sqrt{G_m}\right)\right)U_m^\top\boldsymbol{w} + \left(\sqrt{G_m}\right)^{-1}\sin\left(t\sqrt{G_m}\right)U_m^\top\boldsymbol{b}_1.$$

The overall block Arnoldi approximation is then given by $\boldsymbol{y}_m(t) = U_m\boldsymbol{x}_m(t)$.

We can now apply our framework because the block Krylov subspaces satisfy the nestedness property $A\mathcal{K}_m^{\square}(A,\mathcal{B}) \subseteq \mathcal{K}_{m+1}^{\square}(A,\mathcal{B})$. This yields

$$\boldsymbol{r}_m(t) = -\mathcal{U}_{m+1}\mathcal{g}_{m+1,m}[\boldsymbol{x}_m(t)]_{2m-1:2m},$$

where $\mathcal{g}_{m+1,m} \in \mathbb{R}^{2\times 2}$ denotes the bottom-right block of the block upper Hessenberg matrix arising from the block Arnoldi process. This essentially reproduces the first part of [7, Theorem 3].

EXAMPLE 3.6. In situations where shifted linear systems with A can be efficiently solved, it is often beneficial to work with rational Krylov spaces instead of standard (polynomial)

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Krylov spaces due to the better approximation properties of rational functions compared to polynomials; see, e.g., [25, 26, 27]. A rational Krylov subspace of dimension m with respect to A and $\boldsymbol{b}$ is defined as

$$\mathcal{Q}_m(A, \boldsymbol{b}, q_{m-1}) := q_{m-1}(A)^{-1}\mathcal{K}_m(A, \boldsymbol{b}),$$

with a denominator polynomial $q_{m-1}(z) = \prod_{i=1}^{m-1}(z - z_i)$ with zeros $z_i \notin \sigma(A)$, the so-called *poles* of the rational Krylov space.

Even for nested pole sequences, i.e., $q_m(z) = (z - z_m)q_{m-1}(z)$, we generally have $A\mathcal{Q}_m(A, \boldsymbol{b}, q_{m-1}) \not\subset \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$, so that Theorem 3.1 cannot be applied directly. However, if the pole z_m is chosen "at infinity", i.e., $q_{m-1}(z) = q_m(z)$, then the inclusion $A\mathcal{Q}_m(A, \boldsymbol{b}, q_{m-1}) \subset \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_{m-1}) = \mathcal{Q}_{m+1}(A, \boldsymbol{b}, q_m)$ *does* hold.

This fits quite naturally into the rational Krylov framework. Instead of the Arnoldi relation (2.1), rational Krylov methods yield a relation

$$AU_{m+1}\underline{K}_m = U_{m+1}\underline{G}_m \tag{3.12}$$

with unreduced upper Hessenberg matrices $\underline{G}_m, \underline{K}_m$ of size $(m + 1) \times m$. If $z_m = \infty$, then the last row of $\underline{K}_m$ vanishes, and the relation (3.12) simplifies to

$$AU_m K_m = U_{m+1}\underline{G}_m,$$

where K_m denotes the upper $m \times m$ part of $\underline{K}_m$. In this case, the projection of A onto the rational Krylov space can be computed as

$$U_m^\top A U_m = G_m K_m^{-1},$$

working only with matrices of size $m \times m$; see, e.g., [26, Section 3.1]. Therefore, in order to efficiently evaluate the rational Arnoldi approximation of $f(A)\boldsymbol{b}$, it is common practice in rational Krylov methods to first do a "polynomial step", then form the rational Arnoldi approximation

$$\boldsymbol{f}_m^{\text{rat}} := U_m f(G_m K_m^{-1})\boldsymbol{e}_1\|\boldsymbol{b}\|,$$

and only after that "complete" the rational Arnoldi step by adding a new pole $z_m \neq \infty$ (if the iteration needs to be continued).

Exactly the same approach can be used for computing the rational Krylov residual whenever (3.13) is used in the context of solving an ODE. For ease of exposition we describe the approach for the simplest case, i.e., for solving the IVP (2.4), noting that it is also straightforwardly applicable to all other IVPs. The corresponding rational Arnoldi approximation reads

$$\boldsymbol{y}_m^{\text{rat}}(t) = U_m \exp(-tG_m K_m^{-1})\boldsymbol{e}_1\|\boldsymbol{b}_0\|. \tag{3.13}$$

It is straightforward to verify that the rational Arnoldi approximation (3.13) is characterized by the Galerkin condition

$$\boldsymbol{r}_m^{\text{rat}}(t) \perp \mathcal{Q}_m(A, \boldsymbol{b}, q_{m-1})$$

for the residual $\boldsymbol{r}_m^{\text{rat}}(t) := -(\boldsymbol{y}_m^{\text{rat}})'(t) - A\boldsymbol{y}_m^{\text{rat}}(t)$. Thus, from Theorem 3.1 and Corollary 3.2, we obtain the relation

$$\boldsymbol{r}_m^{\text{rat}}(t) = \beta_m(t)\boldsymbol{u}_{m+1} \qquad \text{with} \qquad \beta_m(t) = -\boldsymbol{h}_{m+1,\bullet}\exp(-tG_m K_m^{-1})\boldsymbol{e}_1\|\boldsymbol{b}_0\|,$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $\boldsymbol{h}_{m+1,\bullet}$ is the last row of the matrix

$$\underline{H}_m := \underline{G}_m K_m^{-1}.$$

We note that further simplifications are not possible because the assumptions for the second assertion of Theorem 3.1 are not fulfilled in general for rational Krylov spaces, unless two consecutive poles are chosen at infinity.

3.4. Bounding the error in terms of the residual. Similar to what we discussed in Section 2.3, the residual norm can be used to obtain a bound for the *error* of the Krylov ODE solution. We note that the Krylov error $\boldsymbol{\xi}_m(t) = \boldsymbol{y}(t) - \boldsymbol{y}_m(t)$ satisfies

$$\begin{aligned}
\boldsymbol{\xi}_m^{(p)}(t) &= \boldsymbol{y}^{(p)}(t) - \boldsymbol{y}_m^{(p)}(t) = -A\boldsymbol{y}(t) + \boldsymbol{w}(t) - \boldsymbol{y}_m^{(p)}(t) \\
&= -A\boldsymbol{y}(t) + \boldsymbol{w}(t) - \boldsymbol{y}_m^{(p)}(t) + A\boldsymbol{y}_m(t) - A\boldsymbol{y}_m(t) \\
&= -A(\boldsymbol{y}(t) - \boldsymbol{y}_m(t)) + \left(-\boldsymbol{y}_m^{(p)}(t) - A\boldsymbol{y}_m(t) + \boldsymbol{w}(t)\right) \\
&= -A\boldsymbol{\xi}_m(t) + \boldsymbol{r}_m(t).
\end{aligned}$$

Therefore, $\boldsymbol{\xi}_m(t)$ can be characterized as the solution of the IVP

$$\begin{cases} \boldsymbol{\xi}_m^{(p)}(t) = -A\boldsymbol{\xi}_m(t) + \boldsymbol{r}_m(t) & \text{on } (0,T], \\ \boldsymbol{\xi}_m(0) = \boldsymbol{\xi}_m'(0) = \cdots = \boldsymbol{\xi}_m^{(p-1)}(0) = \mathbf{0}. \end{cases} \tag{3.14}$$

The solution of (3.14) can be written in terms of a (generalized) variation-of-constants formula

$$\boldsymbol{\xi}_m(t) = \int_0^t K_p(t-\theta, A)\boldsymbol{r}_m(\theta)\,\mathrm{d}\theta, \tag{3.15}$$

where the kernel function $K_p(\tau, A)$ is the inverse Laplace transform (with respect to the variable s) of the function $F(s, A) = (s^p I + A)^{-1}$; see, e.g., [2, Chapter 3]. It follows from [49, Equation (59)] that

$$K_p(\tau, A) = \tau^{p-1} E_{p,p}(-\tau^p A),$$

where $E_{\alpha,\beta}(z) = \sum_{j=0}^{\infty} \frac{z^j}{\Gamma(\alpha j+\beta)}$ is the two-parameter Mittag-Leffler function and $\Gamma(z)$ is the Euler gamma function.

Equation (3.15) directly implies the error bound

$$\begin{aligned}
\|\boldsymbol{\xi}_m(t)\|_* &\le \int_0^t \|K_p(t-\theta, A)\|_* \|\boldsymbol{r}_m(\theta)\|_*\,\mathrm{d}\theta \\
&\le \max_{0\le\theta\le t} \|\boldsymbol{r}_m(\theta)\|_* \int_0^t \|K_p(t-\theta, A)\|_*\,\mathrm{d}\theta.
\end{aligned} \tag{3.16}$$

More concrete error bounds can be obtained for specific combinations of K_p and $\|\cdot\|_*$ by inserting suitable upper bounds for $K_p(\tau, A)$. In the following example, we focus on the two most important special cases $p = 1$ and $p = 2$.

EXAMPLE 3.7. In the first-order case we have $K_1(\tau, A) = \exp(-\tau A)$, and for the Euclidean norm, one can again bound $\|K_1(\tau, A)\| \le C_1 e^{-\tau\omega_1}$, for $\tau \in [0, t]$, to obtain

$$\|\boldsymbol{\xi}_m(t)\| \le \max_{0\le\theta\le t} \|\boldsymbol{r}_m(\theta)\| \cdot C_1 e^{-t\omega_1} \int_0^t e^{-\theta\omega_1}\,\mathrm{d}\theta = C_1 t \varphi_1(-t\omega_1) \max_{0\le\theta\le t} \|\boldsymbol{r}_m(\theta)\|, \tag{3.17}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

exactly reproducing (2.10). An illustration of this error bound is given in Figure 4.1 below.

In the second-order case we have $K_2(\tau, A) = \left(\sqrt{A}\right)^{-1} \sin\left(\tau\sqrt{A}\right)$, and a natural assumption is that A generates an *exponentially bounded cosine family* (see, e.g., [20]), i.e.,

$$\|\cos\left(\tau\sqrt{A}\right)\| \leq C_2 e^{\tau\omega_2},$$

which implies the bound

$$\|\left(\sqrt{A}\right)^{-1} \sin\left(\tau\sqrt{A}\right)\| \leq C_2 \tau \varphi_1(\tau\omega_2),$$

so that (3.16) yields

$$\begin{aligned}\|\boldsymbol{\xi}_m(t)\| &\leq \max_{0\leq\theta\leq t} \|\boldsymbol{r}_m(\theta)\| \cdot C_2 \int_0^t (t-\theta)\varphi_1((t-\theta)\omega_2)\,\mathrm{d}\theta \\ &= C_2 t^2 \varphi_2(\omega_2 t) \max_{0\leq\theta\leq t} \|\boldsymbol{r}_m(\theta)\|,\end{aligned}$$

with the second phi function $\varphi_2(z) = \frac{e^z-1-z}{z^2}$. Specific choices for C_2 and ω_2 again depend on properties of A. For example, when A is Hermitian positive definite, one can take $C_2 = 1$ and $\omega_2 = 0$. For non-Hermitian A, one can either take $C_2 = 1, \omega_2 = \sqrt{\|A\|}$ or $C_2 = \kappa(X)$, $\omega_2 = \max_{\lambda\in\sigma(A)} \Im\left(\sqrt{\lambda}\right)$, where $\kappa(X)$ is the condition number of an eigenvector matrix of A.

EXAMPLE 3.8. We revisit the Poisson system (3.9) considered in Example 3.4. Recall that its solution is given by $\exp(tJQ)\boldsymbol{b}_0$, where Q is symmetric positive definite and J is skew-symmetric. Thus, $\exp(tJQ)$ is unitary with respect to the Q-inner product $\langle\cdot,\cdot\rangle_Q$, which immediately yields $\|\exp(tJQ)\|_Q = 1$, for all $t \geq 0$. We thus directly find the error bound

$$\|\boldsymbol{\xi}_m(t)\|_Q \leq t \cdot \max_{0\leq\theta\leq t} \|\boldsymbol{r}_m(\theta)\|_Q$$

for the approximation produced by the Q-Arnoldi method.

4. Residual-based sketched Krylov methods for ODEs. In this section we discuss in more detail how so-called *sketched Krylov methods*, in particular the sketched Arnoldi (or sFOM) method introduced in [29], fit into the framework of Section 3 and how this enables us to formulate a reliable stopping criterion for the sketched Arnoldi method. We start by briefly recapitulating some important facts about randomized subspace embeddings, which lie at the heart of sketched Krylov methods.

4.1. Randomized sketching. Sketched Krylov methods (and more generally, all methods employing the sketching paradigm) are based on *subspace embeddings* [17, 39, 45, 48], which embed a subspace $\mathcal{V}$ of $\mathbb{R}^n$ into a smaller Euclidean space $\mathbb{R}^s$, $s \ll n$, in a way that distorts norms and inner products in a controlled manner.

For a given subspace $\mathcal{V}$ and distortion parameter $\varepsilon \in [0, 1)$, we call $S \in \mathbb{R}^{s\times n}$ an *ε-subspace embedding for $\mathcal{V}$* if, for all $\boldsymbol{v} \in \mathcal{V}$,

$$(1-\varepsilon)\|\boldsymbol{v}\|^2 \leq \|S\boldsymbol{v}\|^2 \leq (1+\varepsilon)\|\boldsymbol{v}\|^2, \tag{4.1}$$

or equivalently, for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{V}$,

$$\langle\boldsymbol{u},\boldsymbol{v}\rangle - \varepsilon\|\boldsymbol{u}\|\|\boldsymbol{v}\| \leq \langle S\boldsymbol{u}, S\boldsymbol{v}\rangle \leq \langle\boldsymbol{u},\boldsymbol{v}\rangle + \varepsilon\|\boldsymbol{u}\|\|\boldsymbol{v}\|. \tag{4.2}$$

Often, we simply use the established shorthand name *"sketching matrix"* for a subspace embedding S.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

In practice, one typically does not have complete knowledge of the subspace $\mathcal{V}$ that shall be embedded. For example, in the context of the methods discussed in this paper, $\mathcal{V} = \mathcal{K}^{\square}_{m+1}(A, \mathcal{B})$ is a (block) Krylov subspace that has not yet been constructed at the time at which the sketching matrix needs to be fixed.

Therefore, we work with so-called *oblivious embeddings*: For a given subspace dimension d, distortion parameter ε, and failure probability δ, we call $\mathcal{S} \subset \mathbb{R}^{s\times n}$ a family of (ε, δ, d)-oblivious subspace embeddings if, for any subspace $\mathcal{V}$ of dimension d, a randomly drawn $S \in \mathcal{S}$ satisfies property (4.1) and (4.2) with probability $1 - \delta$.

There are many different ways to probabilistically construct families of oblivious subspace embeddings, and we refer the reader to [48] or [39, Section 8] for details. In our experiments reported in Section 5, we always employ a so-called *sparse sign matrix* as the sketching operator, i.e., a matrix of the form

$$S = \sqrt{\frac{n}{s}} \cdot [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n] \in \mathbb{R}^{s\times n},$$

where each column $\boldsymbol{s}_i$ contains exactly $\zeta \in \mathbb{N}$ nonzero entries (at randomly chosen positions) that take the values -1 or 1 with equal probability. Clearly, computing the sketch $S\boldsymbol{v}$ for some vector $\boldsymbol{v} \in \mathbb{R}^n$ is possible in $\mathcal{O}(\zeta n)$ flops. The analysis in [14] shows that sparse sign matrices form a family of (ε, δ, d)-oblivious subspace embeddings if the embedding dimension and the sparsity parameter satisfy $s = \mathcal{O}(\varepsilon^{-2} d \log \frac{d}{\delta})$ and $\zeta = \mathcal{O}(\varepsilon^{-1} d \log \frac{d}{\delta})$, respectively. Empirically, even smaller sparsity parameters appear to work very well in practice. For instance, in [47, Section 3.3], $\zeta = \min\{s, 8\}$ is empirically found to be a very reliable choice, and it is mentioned that there is evidence that an even smaller ζ is often feasible, in particular for large problems. In our experiments reported in Section 5, we go as low as $\zeta = 1$ without observing deterioration of the algorithms.

The quadratic dependence of the sketching dimension on the embedding accuracy, $s = \mathcal{O}(\varepsilon^{-2} d \log \frac{d}{\delta})$, means that one typically needs to accept a rather crude embedding quality ε in order to arrive at a computationally feasible method. A typical choice in many algorithms is to aim for $\varepsilon = 1/\sqrt{2}$, so that $s \sim 2d \log d$.

A viewpoint on the sketching paradigm that is very important in our derivations is that a subspace embedding S induces a (semidefinite) inner product

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_S := \langle S\boldsymbol{u}, S\boldsymbol{v} \rangle \tag{4.3}$$

and a corresponding semi-norm $\|\boldsymbol{v}\|_S = \sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle_S}$. It is easy to see that, due to the embedding property (4.2), $\langle \cdot, \cdot \rangle_S$ is a proper (i.e., positive definite) inner product when restricted to the embedded subspace $\mathcal{V}$; see, e.g., [5, Section 3.1]. Consequently, $\| \cdot \|_S$ is a norm on $\mathcal{V}$. In view of (4.2), one can view $\langle \cdot, \cdot \rangle_S$ and $\| \cdot \|_S$ as slightly distorted versions of the Euclidean inner product and norm, respectively.

4.2. Applying our framework to the sketched Arnoldi method. We now apply our framework to a sketched Arnoldi method, in particular to an implementation of this method employing "basis whitening" that is discussed and analyzed in [29, 42]. In a nutshell, the idea of this method is to cheaply construct a non-orthogonal basis of a Krylov subspace and then retrospectively and implicitly orthogonalize it with respect to the sketched inner product (4.3).

In the following we denote by $V_{m+1} = [\mathcal{V}_1, \ldots, \mathcal{V}_{m+1}]$ an arbitrary (typically non-orthogonal) nested basis of the (block) Krylov subspace $\mathcal{K}^{\square}_{m+1}(A, \mathcal{B})$, where $\mathcal{B} \in \mathbb{R}^{n\times \ell}$, and we assume that $\mathcal{K}^{\square}_1(A, \mathcal{B}) = \operatorname{colspan}\{\mathcal{V}_1\}$ contains all initial conditions as well as $\boldsymbol{w}(t)$ for all $t \in (0, T]$. In the sketched Arnoldi method, we typically obtain V_{m+1} by a k-truncated

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Arnoldi process, which also yields the (block-)k-banded upper Hessenberg matrix $\underline{H}_m$. In the so-called basis whitening step, we compute a thin QR decomposition $SV_{m+1} = Q_{m+1}R_{m+1}$,

$$Q_{m+1} = \begin{bmatrix} Q_m & q_{m+1} \end{bmatrix}, \qquad R_{m+1} = \begin{bmatrix} R_m & r_m \\ 0 & p_{m+1} \end{bmatrix}.$$

Then, $U_{m+1} := V_{m+1}R_{m+1}^{-1}$ is a basis of $\mathcal{K}_{m+1}(A, \mathcal{B})$ that is orthonormal with respect to $\langle \cdot, \cdot \rangle_S$. The *sketched Arnoldi relation* from [43, Proposition 2.1] involving these quantities reads

$$SAU_m = SU_m\left(\widehat{H}_m + c_m e_m^\top\right) + S u_{m+1} p_{m+1} h_{m+1,m} p_m^{-1} e_m^\top, \tag{4.4}$$

where $\widehat{H}_m = R_m H_m R_m^{-1}$, $c_m = r_m h_{m+1,m} p_m^{-1}$, and $e_m \in \mathbb{R}^{m\ell \times \ell}$ contains the last ℓ columns of the identity matrix $I_{m\ell}$.

To apply our framework in the mth step of the algorithm, we choose

$$\underbrace{\mathcal{K}_{m-1}^{\square}(A, \mathcal{B})}_{=:\mathcal{V}_{m-1}} \subseteq \underbrace{\mathcal{K}_{m}^{\square}(A, \mathcal{B})}_{=:\mathcal{V}_{m}} \subseteq \underbrace{\mathcal{K}_{m+1}^{\square}(A, \mathcal{B})}_{=:\mathcal{V}_{m+1}} \subseteq \mathbb{R}^n,$$

define $\boldsymbol{y}_m(t) = U_m \boldsymbol{x}_m(t)$, and demand $\boldsymbol{r}_m(t) \perp_S \mathcal{K}_m^{\square}(A, \mathcal{B})$. As U_{m-1}, U_m, U_{m+1} are orthogonal with respect to $\langle \cdot, \cdot \rangle_S$ and $A\mathcal{K}_j^{\square}(A, \mathcal{B}) \subseteq \mathcal{K}_{j+1}^{\square}(A, \mathcal{B})$, for all $j \in \mathbb{N}$, we have that $\boldsymbol{x}_m(t)$ solves the initial value problem in (3.4) and that we can apply Theorem 3.1 as well as Corollary 3.2. Using (4.4) and noting that $M^* = (SM)^\top S$, we obtain

$$U_m^* A U_m = (SU_m)^\top S A U_m = Q_m^\top S A V_m R_m^{-1} = \widehat{H}_m + c_m e_m^\top \tag{4.5}$$

as well as

$$\|\boldsymbol{r}_m(t)\|_S = \|\boldsymbol{\beta}_m(t)\| \tag{4.6}$$

with

$$\begin{aligned} \boldsymbol{\beta}_m(t) &= -(S\mathcal{U}_{m+1})^\top S A \mathcal{U}_m [\boldsymbol{x}_m(t)]_{m(\ell-1)+1:m\ell} \\ &= -p_{m+1} h_{m+1,m} p_m^{-1} [\boldsymbol{x}_m(t)]_{m(\ell-1)+1:m\ell}. \end{aligned} \tag{4.7}$$

Since $\boldsymbol{r}_m(t) \in \mathcal{K}_{m+1}^{\square}(A, \mathcal{B})$, it follows from the embedding property (4.1) that

$$\|\boldsymbol{r}_m(t)\| \leq \frac{1}{\sqrt{1-\varepsilon}} \|\boldsymbol{r}_m(t)\|_S,$$

i.e., if the sketched residual norm—which we can easily monitor during the sketched Arnoldi method due to (4.6)–(4.7)—is small, then this guarantees that the actual residual norm is also small. Plugging (4.6) into the error bound (3.16) directly yields

$$\|\boldsymbol{\xi}_m(t)\| \leq \frac{1}{\sqrt{1-\varepsilon}} \max_{0 \leq \theta \leq t} \|\boldsymbol{r}_m(\theta)\|_S \int_0^t \|K_p(t-\theta, A)\| \, \mathrm{d}\theta, \tag{4.8}$$

so that, up to a factor $\frac{1}{\sqrt{1-\varepsilon}}$, the same bounds as, e.g., in Example 3.7 can be obtained. For example, in the first-order case and under the usual assumption (2.9), the bound (4.8) turns into

$$\|\boldsymbol{\xi}_m(t)\| \leq \frac{C_1}{\sqrt{1-\varepsilon}} t \varphi_1(-t\omega_1) \max_{0 \leq \theta \leq t} \|\boldsymbol{r}_m(\theta)\|_S, \tag{4.9}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

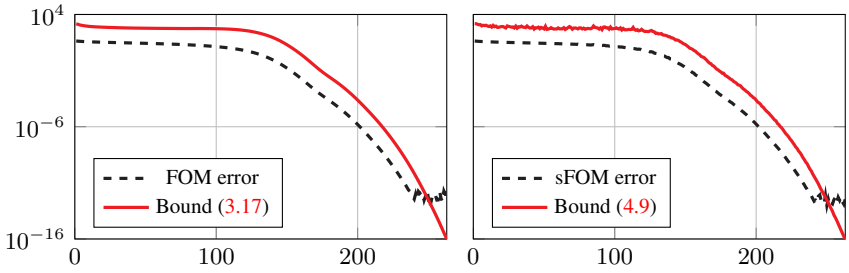

FIG. 4.1. *FOM and sFOM error and error bound for the convection diffusion problem from Section 5.1 for* $N = 100$. *As the spectrum of* A *goes into the left half-plane, we have* $\omega_1 \approx -0.5759$ *leading to* $\varphi(-t\omega_1) \approx 1.3522$. *In sFOM, we orthogonalize each new basis vector against the previous* $k = 2$ *vectors and use a sketching dimension of* $s = 1000$. *As no analytical expression for* ε *is available for the embedding we use, we estimate it empirically, giving* $\varepsilon \approx 0.1351$.

so that a small *sketched residual norm* guarantees that the *actual error norm* of the sketched Arnoldi approximation is small as well. One particular thing to note about the bound (4.9) is that the constants C_1 and ω_1 depend only on spectral properties of the matrix A and not on properties of the sketch-projected matrix, whose spectral region is typically much larger than that of A. Instead, in our bound (4.9), the effect of sketching only enters mildly through the dependence on the embedding quality ε.

We illustrate the bound (4.9) in Figure 4.1 for the model problem considered in Section 5.1, together with the bound obtained for the standard Arnoldi method. We observe that both bounds predict the qualitative behavior of the error very well, although they overestimate the actual magnitude quite substantially—a well-known phenomenon for residual-based error bounds. Both methods and bounds behave almost identically in this experiment. We note for clarity that the bound (4.9) does *not* say anything about how the errors of the sketched and standard Arnoldi approximations compare, though.

4.3. Efficient implementation of the residual-based sketched Arnoldi method. We formulate the sketched Arnoldi method in Algorithm 1. We denote by $\eta_m := \max_{0\le\theta\le t} \|\boldsymbol{r}_m(\theta)\|_S$ the maximum of the sketched residual norm over the integration interval.

Line 1 corresponds to the "block normalization" of $\mathcal{B}$. If $\mathcal{B} = \boldsymbol{b}$ is a vector, then we have $R_{\boldsymbol{b}} = \|\boldsymbol{b}\|$ and thus $V_1 = \boldsymbol{b}/\|\boldsymbol{b}\|$. The update of the QR decomposition in line 6 of Algorithm 1 can be done by any algorithm that iteratively constructs the QR decomposition column by column. We use the modified Gram–Schmidt method with reorthogonalization, as it allows us to work with a thin QR decomposition. Since the matrix Q_m is not needed for any computation in Algorithm 1, one might even use a Q-less QR update.

In line 9 of Algorithm 1, η_m, the maximum of the sketched residual norm, is approximated by computing $\|\boldsymbol{r}_m(\theta_i)\|_S$ over an equidistant grid $0 < \theta_1 = t/n_t < \cdots < \theta_{n_t} = t$, similarly to what is done in [9, 10, 11]. In all experiments reported in Section 5, the residual norm is evaluated at $n_t = 5$ points.

The update of the matrix $\widehat{H}_m$ in line 15 can be done at a cost of $\mathcal{O}(m^2)$, as described in [43, Section 5].

4.4. RT-Restarting for improved stability and efficiency. In our own experiments, as well as in several experiments reported in the literature (see, e.g., [15]), we have observed that for some very hard problems that require many Krylov iterations, the sketched Arnoldi method can become unstable and can stagnate at an unsatisfactory level of accuracy (or even diverges).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 1 Residual-based sketched Arnoldi method.

Input: A, $\mathcal{B}$, t, `tol`, $m_{\max}$
Output: Approximate solution $\boldsymbol{y}_m(t)$

Compute QR decomposition $Q_{\mathcal{B}}R_{\mathcal{B}} = \mathcal{B}$ and set $V_1 \leftarrow \mathcal{B}R_{\mathcal{B}}^{-1}$
Perform truncated Arnoldi step $V_1 \mapsto (V_2, \underline{H}_1)$
Compute sketch $S\mathcal{V}_1$ and QR decomposition $Q_1R_1 = S\mathcal{V}_1$ and set $\widehat{H}_1 \leftarrow H_1$
for $m = 1, \dots, m_{\max}$ **do**
    Compute sketch $S\mathcal{V}_{m+1}$
    Update QR decomposition $(Q_mR_m = SV_m) \mapsto (Q_{m+1}R_{m+1} = SV_{m+1})$
    if `update residual` **then**
        Solve sketch-projected problem (3.4) using (4.5) to obtain $\boldsymbol{x}_m(t)$
        Set $\eta_m \leftarrow \max_{0\le\theta\le t} \|p_{m+1}\hbar_{m+1,m}p_m^{-1}[\boldsymbol{x}_m(\theta)]_{m(\ell-1)+1:m\ell}\|$
        if $\eta_m <$ `tol` **then**
            return $\boldsymbol{y}_m(t) := V_m(R_m^{-1}\boldsymbol{x}_m(t))$
        end if
    end if
    Perform truncated Arnoldi step $(V_{m+1}, \underline{H}_m) \mapsto (V_{m+2}, \underline{H}_{m+1})$
    Update $\widehat{H}_m \mapsto \widehat{H}_{m+1}$
end for
return $\boldsymbol{y}_m(t) := V_m(R_m^{-1}\boldsymbol{x}_m(t))$

A possible remedy is to *restart* the Arnoldi method once stagnation or divergence is observed (e.g., by monitoring whether the residual norm stops decreasing or starts increasing). In the context of solving ODEs, one can do so via sub-time stepping, where the length of the time steps is adaptively determined based on the residual norm. This approach is also known as residual-time (RT) restarting; see [9, 10, 11]. Based on our formulas for the sketched residual norm, this approach, which we briefly recapitulate in the following, can straightforwardly be generalized to the sketched Arnoldi method.

The crucial observation for this approach is that after an arbitrary number m of Krylov steps and for any tolerance $\texttt{tol} > 0$, there always exists a time point $\tau > 0$ such that

$$\max_{0\le\theta\le\tau} \|\boldsymbol{r}_m(\theta)\|_S < \texttt{tol}.$$

After m steps, if the need for a restart is detected, one can thus start a new sketched Arnoldi iteration for the modified IVP

$$\begin{cases} \widetilde{\boldsymbol{y}}^{(p)}(t) = -A\widetilde{\boldsymbol{y}}(t) + \boldsymbol{w}(t+\tau) & \text{on } (0, T-\tau], \\ \widetilde{\boldsymbol{y}}^{(j)}(0) = \boldsymbol{y}_m^{(j)}(\tau), & j = 0, \dots, p-1. \end{cases} \tag{4.10}$$

Clearly, the solution of (4.10) is such that $\widetilde{\boldsymbol{y}}(T-\tau) = \boldsymbol{y}(T)$. Due to the shorter integration interval, problem (4.10) is "easier" to solve than the original IVP, and each subsequent restart cycle reduces the time interval further.

Our experiments suggest that the sketched Arnoldi method is, in fact, very "forgiving" regarding the stabilizing effect of restarts in the sense that no "pre-emptive" restarting is necessary. Even when restarting *after* an instability has been detected, in all our experiments it was still possible to continue iterating to the desired accuracy after restarting.

In our implementation, we therefore simply monitor the slope at which the residual norm decreases by computing $\sigma_m = (\eta_m - \eta_{m-\texttt{update}})/\texttt{update}$, where `update` is a parameter

ETNA
Kent State University and
Johann Radon Institute (RICAM)

that determines after how many iterations we compute the residual norm, setting $\eta_0 := 0$. Often, the residual norm initially stagnates or *increases* in the first few Krylov iterations before steadily decreasing afterward. For detecting the need to restart, we therefore first wait until $\sigma_m < 0$ for the first time. From then on, we continuously monitor whether σ_m exceeds $\texttt{rtol} \cdot \eta_{m-\texttt{update}}$ with some tolerance $\texttt{rtol}$, indicating the need for a restart.

For hard problems that require a large number of Krylov iterations in order to reach the desired accuracy, it might also be attractive to restart the sketched Arnoldi method from time to time in order to keep the computational complexity low. While the cost of the sketched Arnoldi method scales much more favorably with respect to the number of iterations than the cost of the plain Arnoldi method, it still *has* an impact. In particular, evaluating a function at the compressed matrix $\widehat{H}_m + \mathcal{C}_m \mathcal{E}_m^\top$ of size $m\ell$ typically has computational cost $\mathcal{O}\big(m^3\ell^3\big)$. Additionally, a larger number of iterations requires a larger sketching dimension s to guarantee (4.1) with high enough probability. We therefore also provide a maximum number m_{restart} of iterations after which a restart is performed, even when no instability is detected.

A high-level description of this RT-restarting methodology for the sketched Arnoldi method with block size ℓ is given in Algorithm 2.

Algorithm 2 RT-restarting for sketched Arnoldi method.

Input: A, $\mathcal{B}$, t, $\texttt{tol}$, $\texttt{rtol}$, m_{restart}
Output: Approximate solution $\boldsymbol{y}_m(t)$

Set $\texttt{converged} \leftarrow \texttt{false}$
while $\texttt{converged} = \texttt{false}$ **do**
  Run Algorithm 1 with $m_{\max} = m_{\text{restart}}$ or until $\sigma_m > \texttt{rtol} \cdot \eta_{m-\texttt{update}}$
  if $\eta_m < \texttt{tol}$ **then**
    $\texttt{converged} \leftarrow \texttt{true}$
  else
    if $\sigma_m > \texttt{rtol} \cdot \eta_{m-\texttt{update}}$ **then**
      Set $m \leftarrow m - \texttt{update}$
    end if
    Compute τ such that $\max_{0 \le \theta \le \tau} \|\boldsymbol{r}_m(\theta)\|_S < \texttt{tol}$
    Set $\mathcal{B} \leftarrow [\boldsymbol{y}_m(\tau), \boldsymbol{y}_m'(\tau), \ldots, \boldsymbol{y}_m^{(p-1)}(\tau)]$ and $t \leftarrow t - \tau$
  end if
end while
return $\boldsymbol{y}_m(t)$

5. Numerical experiments. In this section we report numerical experiments for realistic test problems in order to illustrate that the sketched Krylov residual norm yields a reliable stopping criterion for the sketched Arnoldi method. Additionally, we compare the performance of the sketched Arnoldi scheme to that of several other established methods, further adding to the existing evidence that this is a very competitive method for computing the action of matrix functions on vectors.

Specifically, we compare our implementation of the residual-based sketched Arnoldi method to the residual-time restarting method from [9, 11][3], to KIOPS (Krylov with In-

[3] Available at https://team.kiam.ru/botchev/expm/. We note that an adaptive version of the method is only implemented for the matrix exponential, and we thus use the non-adaptive version for better comparability.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

complete Orthogonalization Procedure Solver) from [22][4]—which is based on truncated orthogonalization and sub-time stepping driven by a sophisticated error estimate (that is *not* based on the Krylov residual)—as well as the restarted Arnoldi method accelerated by randomized sketching that was recently proposed in [24][5].

For brevity, we denote by

- `FOM`: the standard Arnoldi method (FOM), i.e., (2.2),
- `expRT30`/`phiRT30`: the residual-time restarting methods from [9, 11] with $m_{\text{restart}} = 30$,
- `restart-rand`: the restarted Arnoldi method accelerated by randomized sketching from [24],
- `KIOPS250/500`: KIOPS with $m_{\max} = 250$ or $m_{\max} = 500$ from [22],
- `sFOM250/500`: the sketched Arnoldi method (sFOM) with $m_{\text{restart}} = 250$ or $m_{\text{restart}} = 500$.

Both KIOPS and the sketched Arnoldi method can show significantly better performance if a good approximation for the required Krylov dimension is known in advance. Since this requires a priori knowledge, which cannot be expected in practice, we use the maximum Krylov dimensions $m_{\max} = 250$ or $m_{\max} = 500$ in KIOPS (KIOPS employs a smaller sub-time step if the desired accuracy is not reached after $m_{\max}$ iterations) and the restart lengths $m_{\text{restart}} = 250$ or $m_{\text{restart}} = 500$ in the sketched Arnoldi method across all experiments (although hand-picked values could improve the performance even further).[6] For KIOPS we use the default value $m_{\min} = 10$ for the minimum Krylov dimension.[7]

As mentioned in Section 4.1, we use sparse sign matrices as subspace embeddings in all experiments, with the sparsity parameter chosen as $\zeta = 1$. The sketching dimension is chosen as $s = 2m_{\text{restart}}$, and a truncation parameter of $k = 2$ is used for `sFOM`.[8] For `restart-rand`, the sketching dimension s and restart length m_{restart} are chosen as $s = 160$ and $m_{\text{restart}} = 60$ in Section 5.1 and Section 5.2 and as $s = 400$ and $m_{\text{restart}} = 100$ in Section 5.3, following the choices made by the authors of [24] for their experiments. In all experiments, `expRT30` and `phiRT30` use a restart length of $m_{\max} = 30$ iterations.

In our experiments, we track several different performance indicators for each algorithm: the number of iterations to convergence (m),[9] the CPU time in seconds (`time[s]`), the (sketched) residual norm (η_m) and *relative* error norm (ξ_m^{rel}) at termination, the number of

We have, however, tested both methods for problems involving the exponential, and the results were not substantially different.

[4] Available at https://gitlab.com/stephane.gaudreault/kiops.

[5] A C++ implementation of this method is provided by the authors at https://gitlab.com/nlg550/randomized-krylov. However, in order to allow a meaningful comparison, we instead use a MATLAB implementation which is based on the optimized implementation `funm_kryl` of the restarted Arnoldi method available at https://guettel.com/funm_kryl/, replacing the Arnoldi process by the randomized Arnoldi process from [24, Algorithm 2.2].

[6] Due to the truncated orthogonalization in these algorithms, the restart lengths can be chosen much larger than one typically expects, as essentially the only two limiting factors are the required memory for storing the Krylov basis and the required time for evaluating f at the compressed matrix.

[7] This means that the error estimate is first evaluated after 10 iterations. Then, if the desired accuracy is not yet reached, the number of further Krylov iterations to perform until the next evaluation of the error estimate is adaptively determined by the KIOPS algorithm.

[8] We also ran all experiments using the truncation parameters $k = 1, 4$, and 8. The larger truncation lengths $k = 4, 8$ did not lead to a significant improvement in convergence speed or stability, while substantially increasing the computation time due to the larger number of inner products with long vectors. Using $k = 1$ significantly delayed convergence for the problems considered in Sections 5.2 and 5.3 and showed comparable performance to $k = 2$ for the problem from Section 5.1. We therefore do not include detailed results for these runs and always use the truncation parameter $k = 2$ in the experiments we report.

[9] It should be noted that for `KIOPS` we only track the number of iterations in a sub-step if the sub-step is accepted in order not to artificially inflate the iteration count.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

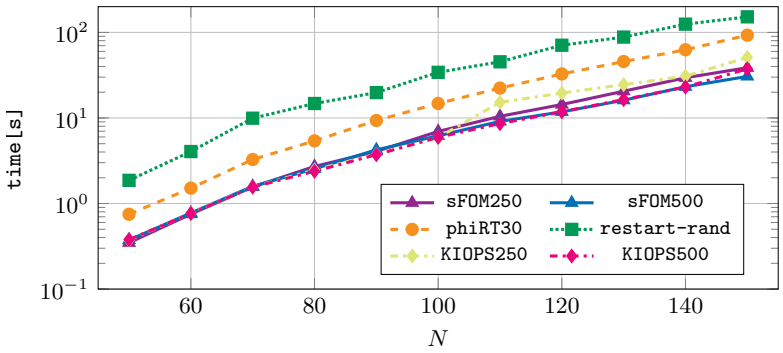

FIG. 5.1. *Execution time for the 3D convection diffusion problem with* $T = 1$.

matrix-vector products with A (`mvecs`), the number of inner products with vectors of size n (`n-prods`) and size s (`s-prods`), the number of matrix function evaluations (`mfuns`), the size of the largest matrix function that needs to be evaluated (`mfunsize`), and the number of other operations with vectors of length n (`n-ops`). `FOM` was not benchmarked in this way due to its high execution time and is only used as a baseline for comparison of the convergence curves. The relative error norms are computed as

$$\xi_m^{\text{rel}} := \frac{||\boldsymbol{y}_{\text{ref}}(T) - \boldsymbol{y}_m(T)||}{||\boldsymbol{y}_{\text{ref}}(T)||},$$

where the reference solution $\boldsymbol{y}_{\text{ref}}(T)$ was pre-computed using `FOM` with a required residual norm of $\eta_m \leq 10^{-15}$.

In the implementation of `restart-rand` and `sFOM`, we use the MATLAB function `expmv`, which implements the algorithm proposed in [1] and allows one to efficiently compute the action of the matrix exponential on a vector over equally-spaced time intervals by reusing some computations.

All algorithms are run until $\eta_m \leq \texttt{tol} = 10^{-8}$. For `KIOPS` (which is not a residual-based method), this is enforced by simply testing if $\eta_m \leq \texttt{tol}$ after termination and supplying a smaller tolerance for the error estimate if $\eta_m > \texttt{tol}$.

All experiments are carried out in MATLAB 25.1 (R2025a) on a PC with an Intel Core Ultra 7 165U CPU with a clock rate of 4.9 GHz and 32 GB RAM. The MATLAB code can be found at https://github.com/kriegerbuw/ks-res-ode.

5.1. Three-dimensional convection-diffusion equation. As a first model problem, we consider the semi-discretization of the three-dimensional convection-diffusion equation

$$u_t = -\nu \Delta u + w \nabla u$$

on $[0, T] \times [0, 1]^3$ by centered finite differences with N grid points in each spatial dimension. The convection field is chosen as $w = (x \sin(x), y \cos(y), e^{z^2 - 1})$, and the diffusion coefficient is $\nu = 5 \cdot 10^{-3}$. We add a constant inhomogeneity $\boldsymbol{g}$, which is a vector containing the values of the function $10e^{-100((x-\frac{1}{2})^2+(y-\frac{1}{2})^2+(z-\frac{1}{2})^2)}$ evaluated at the grid points, and choose the initial value $\boldsymbol{b}_0$ as a vector with normally distributed entries, normalized to have norm 1. This leads to the ODE IVP

$$\begin{cases} \boldsymbol{y}'(t) = -A\boldsymbol{y}(t) + \boldsymbol{g} & \text{on } (0, T], \\ \boldsymbol{y}(0) = \boldsymbol{b}_0, \end{cases}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 5.1
Performance indicators for the 3D convection diffusion problem with $N = 150$ and $T = 1$.

	phiRT30	restart-rand	KIOPS250	KIOPS500	sFOM250	sFOM500
m	644	420	500	356	400	350
time[s]	92.1598	153.9566	43.5597	32.5509	34.5991	28.0607
η_m	$5.3856 \cdot 10^{-9}$	$9.2843 \cdot 10^{-13}$	$2.7525 \cdot 10^{-17}$	$6.4268 \cdot 10^{-9}$	$2.4726 \cdot 10^{-9}$	$7.7469 \cdot 10^{-10}$
ξ_m^{rel}	$6.4729 \cdot 10^{-12}$	$4.8547 \cdot 10^{-14}$	$1.1516 \cdot 10^{-14}$	$6.666 \cdot 10^{-13}$	$2.2995 \cdot 10^{-13}$	$7.3458 \cdot 10^{-14}$
mvecs	644	420	500	356	402	351
s-prods	0	12811	0	0	85404	122852
n-prods	10536	0	1502	1069	800	700
mfuns	1330	14	30	26	17	14
mfunsize	31	420	251	357	250	350
n-ops	10536	13170	1498	1067	1600	1400

with solution

$$\boldsymbol{y}(t) = \exp(-tA)\boldsymbol{b} + t\varphi_1(-tA)\boldsymbol{g} = t\varphi_1(-tA)(\boldsymbol{g} - A\boldsymbol{b}) + \boldsymbol{b}.$$

In our tests, we vary N between 50 and 150, leading to matrices of size between $n = 125\,000$ and $n = 3\,375\,000$.

Figure 5.1 displays the CPU times required by the different algorithms to reach the desired accuracy at $T = 1$ for all considered problem instances. Across all values of N, sFOM500 is competitive with KIOPS500, the otherwise fastest tested algorithm. At $N = 120$, sFOM250 starts to become a bit slower than KIOPS500 and sFOM500, as it has to restart. However, we can see that it is still competitive with KIOPS250. For $N = 110$, KIOPS250 has to employ time-stepping for the first time while sFOM still converges in 250 iterations or less, leading to a speedup of about 1.67. It should, however, be noted that after time-stepping, KIOPS250 always performs the full 250 iterations, and sFOM250 therefore becomes only marginally faster than KIOPS250 as its number of iterations approaches 500 for the larger problems. All versions of KIOPS and sFOM are significantly faster than phiRT30 and restart-rand for this problem.

To explain the observed CPU times, we take an exemplary look at the performance indicators measured for $N = 150$, which can be found in Table 5.1 (the results look very similar across all N, so we only report them for the largest value). We observe that the performance indicators for KIOPS500 and sFOM500 look very similar, which leads to similar CPU times; the only real difference being the large number of (very cheap) inner products with vectors of size s, which is somewhat counteracted by the use of fewer (expensive) inner products with vectors of size n. KIOPS250 and sFOM250 take slightly longer as they need more iterations to converge and have to restart once. Although phiRT30 requires more matrix function evaluations as well as more inner products and other vector operations of size n than KIOPS and sFOM, its cheap matrix function evaluations (due to extremely small compressed matrices of size at most $m = 30$) may make it competitive with KIOPS and sFOM once they have to restart a couple of times, as their restarts are much more costly. Lastly, restart-rand has the highest CPU time across all grid sizes due to the high number of operations with vectors of length n that arise in the explicit orthogonalization of the basis with respect to $\langle \cdot, \cdot \rangle_S$ and the high cost associated with the matrix function evaluations in later iterations.

Figure 5.2 displays the convergence curves for the different algorithms for $N = 150$. We observe that the sketched residual in sFOM500 behaves almost identically to the residual of FOM, while KIOPS500 shows a slightly delayed convergence as it is essentially the truncated Arnoldi algorithm at this point. Similar behavior can be observed for sFOM250 and KIOPS250, where convergence after restarting accelerates quickly for sFOM250 and a bit

ETNA
Kent State University and
Johann Radon Institute (RICAM)

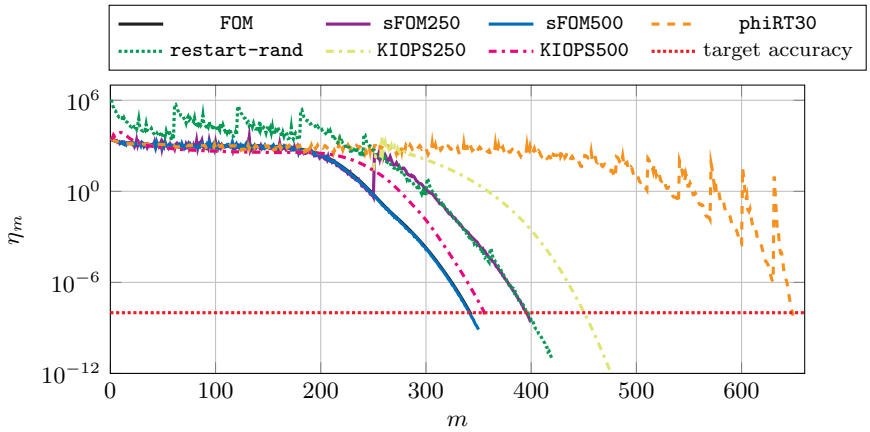

FIG. 5.2. *Convergence curves for the 3D convection diffusion problem considered in Section 5.1 with* $N = 150$ *and* $T = 1$.

more slowly for `KIOPS250`. Due to its longer restarts, `restart-rand` converges in fewer iterations than `phiRT30`, although its execution time is higher.

5.2. Photonic crystal—three-dimensional Maxwell equations. Our next model problem is taken from [9, Section 3.2] and was originally considered in [36]. We consider a semi-discretization of the three-dimensional Maxwell equations

$$H_t = -\frac{1}{\mu_0}\nabla E, \qquad E_t = -\frac{1}{\epsilon_0}\nabla H$$

in a lossless, source-free medium. Here, H denotes the magnetic field, and E denotes the electric field, and the permeability coefficient μ_0 and permittivity coefficient ϵ_0 depend on the spatial position. The setup of the domain and the boundary conditions are chosen exactly as in [9]: We consider the spatial domain $\Omega = [-6.05, 6.05]^3$, which is filled with air (corresponding to a relative permittivity $\epsilon_0 = 1$). Inside this domain, there is a dielectric material (with relative permittivity $\epsilon_0 = 5$) in the region $\Omega_D = [-4.55, 4.55]^3$, which is pierced by 27 air-filled cylinders with radius 1.4 and centers $(x_i, y_j, z_k) = (3.03i, 3.03j, 3.03k)$, where $i, j, k \in \{-1, 0, 1\}$. The permeability is $\mu_0 \equiv 1$ across the whole domain Ω. We consider a perfectly conducting domain boundary, i.e., the boundary conditions are chosen such that the tangential electric field is zero. As initial values, we consider an all-zero magnetic field H and a point light source located in the middle of the spatial domain as the only nonzero value in E.

For the spatial discretization, we employ a finite-difference staggered Yee discretization on an $80 \times 80 \times 80$ grid (leading to a system of size $n = 3\,188\,646$). We seek a solution at final time $T = 10$. We obtain an ODE IVP

$$\begin{cases} \boldsymbol{y}'(t) = -A\boldsymbol{y}(t) & \text{on } (0, T], \\ \boldsymbol{y}(0) = \boldsymbol{b}_0, \end{cases}$$

with solution

$$\boldsymbol{y}(t) = \exp(-tA)\boldsymbol{b}_0.$$

Table 5.2 depicts the performance indicators, and Figure 5.3 displays the convergence curves for this experiment. Due to the reasons already mentioned in Section 5.1, both versions

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 5.2
Performance indicators for the photonic crystal problem on a $80 \times 80 \times 80$ *grid with* $T = 10$.

	expRT30	restart-rand	KIOPS250	KIOPS500	sFOM250	sFOM500
m	735	300	500	296	325	300
time[s]	91.09	100.3193	34.2831	19.5276	22.1187	18.0735
η_m	$1.86 \cdot 10^{-9}$	$8.0679 \cdot 10^{-15}$	$2.0706 \cdot 10^{-132}$	$1.4911 \cdot 10^{-13}$	$2.9242 \cdot 10^{-25}$	$5.7778 \cdot 10^{-16}$
ξ_m^{rel}	$8.95 \cdot 10^{-12}$	$9.5395 \cdot 10^{-13}$	$3.7886 \cdot 10^{-14}$	$3.7818 \cdot 10^{-14}$	$5.947 \cdot 10^{-12}$	$1.7043 \cdot 10^{-13}$
mvecs	735	300	500	296	325	300
s-prods	0	9151	0	0	68454	90302
n-prods	12015	0	1502	889	650	600
mfuns	4434	10	28	26	14	12
mfunsize	30	300	251	297	250	300
n-ops	12015	9390	1498	887	1296	1198

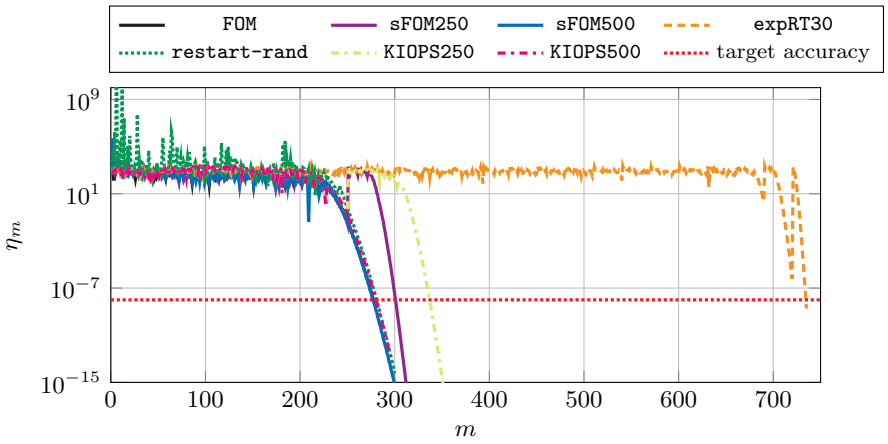

FIG. 5.3. *Convergence curves for the photonic crystal problem on a* $80 \times 80 \times 80$ *grid with* $T = 10$.

of sFOM are competitive with KIOPS500, while KIOPS250 dramatically overshoots the target accuracy as it only tests whether the target accuracy is reached at the end of every restart. expRT30 and restart-rand are significantly slower than KIOPS and sFOM, which is mostly caused by the high number of operations with vectors of size n, with expRT30 additionally requiring almost twice as many matrix-vector products as the other algorithms due to its delayed convergence.

5.3. Vibrating membrane—two-dimensional wave equation. In this experiment, taken from [24, Section 5.2], we simulate the vibrations of a circular membrane Ω of radius $r = 1$ that is attached to a rigid frame. At a time point t, the height of the membrane at a point $(x, y) \in \Omega$ is described by the solution u of the wave equation

$$u_{tt} = -\nu^2 \Delta u,$$

where $\nu > 0$ is the speed at which transversal waves propagate in the membrane. Using the same setup as in [24], we set the initial conditions to

$$u(x, y, 0) = J_4\left(\eta_4 \sqrt{x^2 + y^2}\right), \qquad u_t(x, y, 0) = 0,$$

where J_4 is the fourth-order Bessel function of the first kind and η_4 is the fourth positive root of J_4.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 5.3
Performance indicators for the vibrating membrane problem with $T = 1.5$.

	expRT30	restart-rand	sFOM250	sFOM500
m	9765	1300	1850	1950
time[s]	504.178	341.9677	93.8612	110.766
η_m	$7.2574 \cdot 10^{-9}$	$6.7072 \cdot 10^{-38}$	$3.8647 \cdot 10^{-47}$	$7.6118 \cdot 10^{-19}$
ξ_m^{rel}	$3.436 \cdot 10^{-10}$	$4.8321 \cdot 10^{-9}$	$3.4293 \cdot 10^{-10}$	$3.3979 \cdot 10^{-10}$
mvecs	9765	1300	3550	3400
s-prods	0	65651	869392	1666318
n-prods	161010	0	23545	21277
mfuns	59176	26	137	168
mfunsize	30	1300	250	500
n-ops	161010	66850	20721	19487

We semi-discretize the wave equation using a P1-finite element method (FEM), leading to an ODE IVP

$$\begin{cases} \boldsymbol{y}''(t) = -\nu^2 L \boldsymbol{y}(t) & \text{on } (0, T], \\ \boldsymbol{y}(0) = \boldsymbol{b}_0, \quad \boldsymbol{y}'(0) = \boldsymbol{0}, \end{cases} \tag{5.1}$$

with $L = M^{-1}A$, where A denotes the stiffness matrix and M is the lumped mass matrix and where the initial conditions are given by $[\boldsymbol{b}_0]_k = J_4(\eta_4\sqrt{x_k^2 + y_k^2})$, with (x_k, y_k) being the coordinates of the kth node in the FEM mesh. When using a P1-finite element of size $2.5 \cdot 10^{-3}$, this leads to a matrix of size $n = 580\,012$. The solution to (5.1) is

$$\boldsymbol{y}(t) = \cos\left(\nu t\sqrt{L}\right)\boldsymbol{b}_0.$$

For `expRT30` and `KIOPS`, which are specifically designed to solve first-order ODEs, we transform (5.1) to a first-order problem by defining

$$\boldsymbol{z}(t) = \begin{bmatrix} \boldsymbol{y}(t) \\ \boldsymbol{y}'(t) \end{bmatrix}, \qquad \mathcal{L} = \begin{bmatrix} 0 & -I \\ \nu^2 L & 0 \end{bmatrix}, \qquad \boldsymbol{z}_0 = \begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{0} \end{bmatrix}.$$

This yields

$$\begin{cases} \boldsymbol{z}'(t) = -\mathcal{L}\boldsymbol{z}(t) & \text{on } (0, T], \\ \boldsymbol{z}(0) = \boldsymbol{z}_0, \end{cases} \tag{5.2}$$

with solution

$$\boldsymbol{z}(t) = \exp(-t\mathcal{L})\boldsymbol{z}_0.$$

Note that while this first-order formulation requires working with vectors of length $2n$ (which, e.g., increases the orthogonalization cost and the memory required to store the Krylov basis), matrix-vector products with $\mathcal{L}$ can be performed at essentially the same cost as those with L via

$$\begin{bmatrix} \boldsymbol{v}^{(1)} \\ \boldsymbol{v}^{(2)} \end{bmatrix} \mapsto \begin{bmatrix} -\boldsymbol{v}^{(2)} \\ \nu^2 L \boldsymbol{v}^{(1)} \end{bmatrix}.$$

We also use the equivalent first-order formulation (5.2) for solving the projected IVP in `restart-rand` and `sFOM`, as this led to improved stability in preliminary experiments.

Table 5.3 illustrates the performance indicators, and Figure 5.4 displays the convergence curves from our experiments, where we choose $\nu = 0.85$ and $T = 1.5$. Note that since

ETNA
Kent State University and
Johann Radon Institute (RICAM)

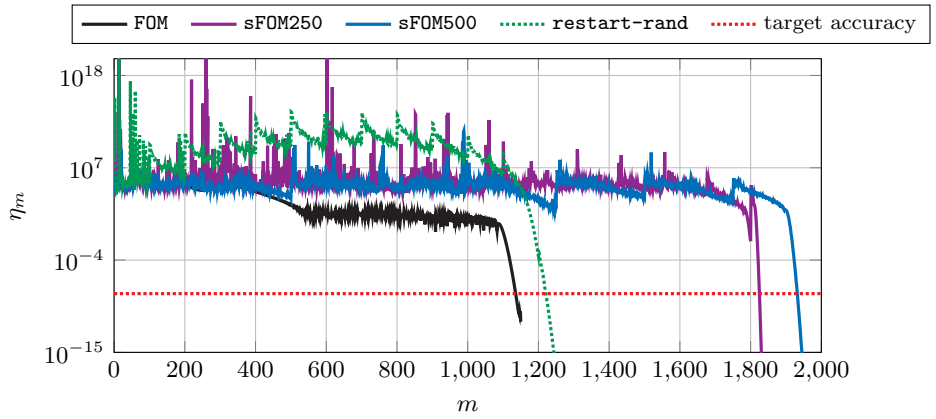

FIG. 5.4. *Convergence curves for the vibrating membrane problem with* $T = 1.5$.

both KIOPS250 and KIOPS500 failed to converge, they are not listed in the table[10], and expRT30 is not plotted as it behaves very similarly to the other experiments but makes Figure 5.4 substantially less readable. For the second-order problem, sFOM was up to five times as fast as expRT30 and restart-rand. It should be noted that when doing an RT-restart with time step τ, we in general have $\boldsymbol{u}_m'(\tau) \neq \mathbf{0}$. Therefore, after restarting, sFOM employs a block Krylov approach (cf. Example 3.5). This increases mvecs by a factor of two as well as s-prods and n-prods by a factor of four after the first restart but has a much milder impact on the computation time, as all corresponding operations are blocked. Due to the high number of iterations, restart-rand has to evaluate functions of several comparatively large matrices, up to a matrix dimension of 1200, which becomes extremely costly in addition to the high number of n-ops.

We also see that sFOM500 takes more iterations to converge than sFOM250, leading to very similar numbers in most performance indicators and sFOM250 being about 20 seconds faster. The reason for this is that in their first (non-blocked) restarts, both algorithms are able to reduce the residual to just slightly above the desired tolerance of 10^{-8} for 30–40% of the time interval but only actually reach the tolerance for $< 1\%$ of the interval, leading to extremely costly first restarts for both. After the restart, however, sFOM500 converges slightly faster than sFOM250, due to the fact that it does not need the entire 250 iterations that are "lost" in the first restart for sFOM250.

6. Conclusions and outlook. We have introduced a general framework that specifies conditions under which the Krylov ODE residual is easily and efficiently computable, which encompasses both polynomial and rational Krylov methods employing arbitrary inner products. As a particularly important special case that is covered by our framework, we have discussed Krylov methods with randomized sketching in depth. In particular, our framework allows us to derive a reliable, rigorous stopping criterion for the sketched Arnoldi method in the context of solving ODEs, while, so far, only heuristic stopping criteria have been available in the literature. Additionally, we have incorporated residual-time restarting into the sketched Arnoldi method to make it more stable and efficient. Numerical experiments on large-scale real-world problems demonstrate the use of the sketched residual as a stopping criterion for

[10] KIOPS was also tested with other maximum Krylov dimensions—up to $m_{\max} = 1500$—of which all either produced NaNs or reduced the time interval to below machine precision.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

the sketched Arnoldi method as well as the competitiveness of the sketched Arnoldi scheme with several state-of-the-art methods.

As a future research goal, it would be valuable to derive rigorous relations between the residuals of the sketched Arnoldi method and the standard (full) Arnoldi method. This would allow us to more directly characterize possible delays of convergence introduced by sketching. Another interesting area for future research is the extension of our methodology to *fractional* differential equations, whose solutions can be expressed in terms of Mittag-Leffler functions [31].

Acknowledgment. We are indebted to Mike Botchev and Leonid Knizhnerman for providing us with the matrix and initial conditions used in the experiment in Section 5.2.

REFERENCES

[1] 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 (2011), pp. 488–511.
[2] W. ARENDT, C. J. K. BATTY, M. HIEBER, AND F. NEUBRANDER, *Vector-Valued Laplace Transforms and Cauchy Problems*, 2nd ed., Birkhäuser/Springer, Basel, 2011.
[3] W. E. ARNOLDI, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
[4] O. BALABANOV AND L. GRIGORI, *Randomized Gram-Schmidt process with application to GMRES*, SIAM J. Sci. Comput., 44 (2022), pp. A1450–A1474.
[5] O. BALABANOV AND A. NOUY, *Randomized linear algebra for model reduction. Part I: Galerkin methods and error estimation*, Adv. Comput. Math., 45 (2019), pp. 2969–3019.
[6] B. BECKERMANN AND L. REICHEL, *Error estimates and evaluation of matrix functions via the Faber transform*, SIAM J. Numer. Anal., 47 (2009), pp. 3849–3883.
[7] M. A. BOTCHEV, *A block Krylov subspace time-exact solution method for linear ordinary differential equation systems*, Numer. Linear Algebra Appl., 20 (2013), pp. 557–574.
[8] M. A. BOTCHEV, V. GRIMM, AND M. HOCHBRUCK, *Residual, restarting, and Richardson iteration for the matrix exponential*, SIAM J. Sci. Comput., 35 (2013), pp. A1376–A1397.
[9] M. A. BOTCHEV AND L. KNIZHNERMAN, *ART: adaptive residual-time restarting for Krylov subspace matrix exponential evaluations*, J. Comput. Appl. Math., 364 (2020), Paper No. 112311, 14 pages.
[10] M. A. BOTCHEV, L. KNIZHNERMAN, AND M. SCHWEITZER, *Krylov subspace residual and restarting for certain second order differential equations*, SIAM J. Sci. Comput., 46 (2024), pp. S223–S253.
[11] M. A. BOTCHEV, L. KNIZHNERMAN, AND E. E. TYRTYSHNIKOV, *Residual and restarting in Krylov subspace evaluation of the φ function*, SIAM J. Sci. Comput., 43 (2021), pp. A3733–A3759.
[12] L. BURKE, S. GÜTTEL, AND K. M. SOODHALTER, *GMRES with randomized sketching and deflated restarting*, SIAM J. Matrix Anal. Appl., 46 (2025), pp. 702–725.
[13] E. CELLEDONI AND I. MORET, *A Krylov projection method for systems of ODEs*, Appl. Numer. Math., 24 (1997), pp. 365–378.
[14] M. B. COHEN, *Nearly tight oblivious subspace embeddings by trace inequalities*, in Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, R. Kraughgamer, ed., ACM, New York, 2016, pp. 278–287.
[15] A. CORTINOVIS, D. KRESSNER, AND Y. NAKATSUKASA, *Speeding up Krylov subspace methods for computing $f(A)b$ via randomization*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 619–633.
[16] M. CROUZEIX AND C. PALENCIA, *The numerical range is a $(1+\sqrt{2})$-spectral set*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 649–655.
[17] P. DRINEAS, M. W. MAHONEY, AND S. MUTHUKRISHNAN, *Subspace sampling and relative-error matrix approximation: column-based methods*, in Algorithms—ESA 2006, Y. Azar and T. Erlebach, eds., Lecture Notes in Comput. Sci., 4168, Springer, Berlin, 2006, pp. 304–314.
[18] V. L. DRUSKIN AND L. KNIZHNERMAN, *Two polynomial methods for calculating functions of symmetric matrices*, U.S.S.R. Comput. Math. Math. Phys., 29 (1989), pp. 112–121.
[19] M. FASI, S. GAUDREAULT, K. LUND, AND M. SCHWEITZER, *Challenges in computing matrix functions*, Preprint on arXiv, 2024. https://arxiv.org/abs/2401.16132
[20] H. O. FATTORINI, *Second Order Linear Differential Equations in Banach Spaces*, North-Holland, Amsterdam, 1985.
[21] A. FROMMER, S. GÜTTEL, AND M. SCHWEITZER, *Efficient and stable Arnoldi restarts for matrix functions based on quadrature*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 661–683.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[22] S. GAUDREAULT, G. RAINWATER, AND M. TOKMAN, *KIOPS: a fast adaptive Krylov subspace solver for exponential integrators*, J. Comput. Phys., 372 (2018), pp. 236–255.
[23] W. GAUTSCHI, *Numerical integration of ordinary differential equations based on trigonometric polynomials*, Numer. Math., 3 (1961), pp. 381–397.
[24] N. L. GUIDOTTI, P.-G. MARTINSSON, J. A. ACEBRÓN, AND J. MONTEIRO, *Accelerating a restarted Krylov method for matrix functions with randomization*, Preprint on arXiv, 2025. https://arxiv.org/abs/2503.22631
[25] S. GÜTTEL, *Rational Krylov Methods for Operator Functions*, PhD. Thesis, Fakultät für Mathematik und Informatik der Technischen Universität Bergakademie Freiberg, Freiberg, 2010.
[26] ———, *Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection*, GAMM-Mitt., 36 (2013), pp. 8–31.
[27] S. GÜTTEL AND L. KNIZHNERMAN, *A black-box rational Arnoldi variant for Cauchy-Stieltjes matrix functions*, BIT, 53 (2013), pp. 595–616.
[28] S. GÜTTEL, D. KRESSNER, AND K. LUND, *Limited-memory polynomial methods for large-scale matrix functions*, GAMM-Mitt., 43 (2020), Paper No. e202000019, 19 pages.
[29] S. GÜTTEL AND M. SCHWEITZER, *Randomized sketching for Krylov approximations of large-scale matrix functions*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 1073–1095.
[30] S. GÜTTEL AND I. SIMUNEC, *A sketch-and-select Arnoldi process*, SIAM J. Sci. Comput., 46 (2024), pp. A2774–A2797.
[31] H. J. HAUBOLD, A. M. MATHAI, AND R. K. SAXENA, *Mittag-Leffler functions and their applications*, J. Appl. Math., (2011) 2011, Paper No. 298628, 51 pages.
[32] N. J. HIGHAM, *Functions of Matrices*, SIAM, Philadelphia, 2008.
[33] N. J. HIGHAM AND A. H. AL-MOHY, *Computing matrix functions*, Acta Numer., 19 (2010), pp. 159–208.
[34] M. HOCHBRUCK AND C. LUBICH, *A Gautschi-type method for oscillatory second-order differential equations*, Numer. Math., 83 (1999), pp. 403–426.
[35] M. HOCHBRUCK AND A. OSTERMANN, *Exponential integrators*, Acta Numer., 19 (2010), pp. 209–286.
[36] J. KOLE, M. T. FIGGE, AND H. DE RAEDT, *Unconditionally stable algorithms to solve the time-dependent Maxwell equations*, Phys. Rev. E, 64 (2001), Paper No. 066705, 14 pages.
[37] A. KOSKELA, *Approximating the matrix exponential of an advection-diffusion operator using the incomplete orthogonalization method*, in Numerical Mathematics and Advanced Applications—ENUMATH 2013, A. Abdulle, S. Deparis, D. Kressner, F. Nobile, and M. Picasso, eds., Lect. Notes Comput. Sci. Eng., 103, Springer, Cham, 2015, pp. 345–353.
[38] S. MAIER, N. MARHEINEKE, AND A. FROMMER, *Energy-preserving iteration schemes for Gauss collocation integrators*, Linear Algebra Appl., 2025, published online. https://doi.org/10.1016/j.laa.2025.09.005
[39] P.-G. MARTINSSON AND J. A. TROPP, *Randomized numerical linear algebra: foundations and algorithms*, Acta Numer., 29 (2020), pp. 403–572.
[40] Y. NAKATSUKASA AND J. A. TROPP, *Fast and accurate randomized algorithms for linear systems and eigenvalue problems*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 1183–1214.
[41] J. NIESEN AND W. M. WRIGHT, *Algorithm 919: a Krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators*, ACM Trans. Math. Software, 38 (2012), Paper No. 22, 19 pages.
[42] D. PALITTA, M. SCHWEITZER, AND V. SIMONCINI, *Sketched and truncated polynomial Krylov methods: evaluation of matrix functions*, Numer. Linear Algebra Appl., 32 (2025), Paper No. e2596, 16 pages.
[43] ———, *Sketched and truncated polynomial Krylov subspace methods: matrix Sylvester equations*, Math. Comp., 94 (2025), pp. 1761–1792.
[44] Y. SAAD, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.
[45] T. SARLÓS, *Improved approximation algorithms for large matrices via random projections*, in 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), IEEE Conference Proceedings, Los Alamitos, 2006, pp. 143–152.
[46] R. B. SIDJE, *Expokit: a software package for computing matrix exponentials*, ACM Trans. Math. Software, 24 (1998), pp. 130–156.
[47] J. A. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Streaming low-rank matrix approximation with an application to scientific simulation*, SIAM J. Sci. Comput., 41 (2019), pp. A2430–A2463.
[48] D. P. WOODRUFF, *Sketching as a tool for numerical linear algebra*, Found. Trends Theor. Comput. Sci., 10 (2014), pp.1–157.
[49] P. VAN MIEGHEM, *The Mittag-Leffler function*, Preprint on arXiv, 2020. https://arxiv.org/abs/2005.13330