

SKETCH-AND-RESTART: RANDOMIZED SKETCHING IN QUADRATURE-BASED RESTARTING FOR MATRIX FUNCTIONS

STEFAN GÜTTEL*, JINGYU LIU†, AND LAURI NYMAN‡

Abstract. We develop a sketch-and-restart framework for computing the action of a matrix function on a vector, $f(\mathbf{A})\mathbf{b}$, where $\mathbf{A}$ is large, sparse, and non-Hermitian. The framework combines quadrature-based restarting with Arnoldi-like decompositions generated by sketched or truncated Arnoldi processes. Within this framework, we develop two classes of restarted algorithms. The first uses a fixed Krylov subspace dimension and is based either on the sketched Arnoldi process or on a new sketched harmonic Arnoldi process proposed in this work. The second class chooses the Krylov subspace dimension adaptively by running the truncated Arnoldi process until the condition number of the generated basis, estimated from its sketch, exceeds a prescribed threshold. We also establish the convergence of the restarted sketched harmonic Arnoldi method for Stieltjes functions under the assumption that $\mathbf{A}$ is positive real. Numerical experiments demonstrate the effectiveness of the proposed framework, including the computational savings achieved through sketching, the storage reduction enabled by adaptive truncation, and the acceleration obtained from thick restarting.

Key words. matrix function, Krylov method, sketching, restarting

AMS subject classifications. 65F60, 65F50, 68W20

1. Introduction. This paper considers the computation of $f(\mathbf{A})\mathbf{b}$, where $\mathbf{A} \in \mathbb{C}^{N \times N}$ is a matrix, $\mathbf{b} \in \mathbb{C}^N$ is a vector, and f is a suitable scalar function. Such computations arise in many areas of scientific computing.

If $\mathbf{A}$ is small and dense, one can first compute $f(\mathbf{A})$ explicitly using algorithms such as those described in [14], and then apply it to $\mathbf{b}$. For large matrices, however, this approach is infeasible because of its computational and memory costs. *Krylov subspace methods* are therefore commonly used to approximate $f(\mathbf{A})\mathbf{b}$, as they require only matrix-vector products with $\mathbf{A}$, which can often be computed efficiently when $\mathbf{A}$ is sparse. Their main limitation is the need to store and orthogonalize the Krylov basis. For non-Hermitian matrices, the Arnoldi process requires $\mathcal{O}(Nm^2)$ operations to generate m orthonormal basis vectors and $\mathcal{O}(Nm)$ storage for the basis, and these costs can become substantial when m is large.

Many approaches have been proposed to address these limitations. One important class is based on *restarting*, in which only a fixed number of basis vectors need to be stored [1, 8]. The *quadrature-based restarting* method of [11] uses an integral representation of f , together with interpolation polynomials, to derive an integral representation of the error. Approximating this error representation by quadrature allows the process to be restarted without storing all previously generated basis vectors. However, restarted methods may converge slowly, and this issue can be alleviated by *thick restarting* strategies [4, 9, 16].

Another class of approaches is based on *randomized sketching* [5, 13]. These methods approximate $f(\mathbf{A})\mathbf{b}$ using a nonorthonormal basis of a Krylov subspace, often generated by the sketched Arnoldi process [2] or the truncated Arnoldi process [17]. The sketched Arnoldi process computes the inner products required for orthogonaliza-

*Department of Mathematics, The University of Manchester, Oxford Road, Manchester, M139PL, United Kingdom, stefan.guettel@manchester.ac.uk

†School of Mathematical Sciences, Fudan University, 220 Handan Road, Shanghai, 200433, China, jyliu22@m.fudan.edu.cn

‡Department of Mathematics, The University of Manchester, Oxford Road, Manchester, M139PL, United Kingdom, lauri.nyman@manchester.ac.uk

tion in the sketched space, thereby reducing the orthogonalization cost by nearly one half. The truncated Arnoldi process reduces the orthogonalization cost to $\mathcal{O}(Nmt)$, where t is the truncation parameter. However, the condition number of the generated basis may grow exponentially [3], which limits the usable Krylov subspace dimension.

We focus on the case in which $\mathbf{A}$ is large, sparse, and non-Hermitian. We develop a sketch-and-restart framework for computing $f(\mathbf{A})\mathbf{b}$ by combining randomized sketching with quadrature-based restarting. The key observation is that several Krylov and sketched Krylov approximations admit the common form $\mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta$, which is equivalent to a polynomial approximation $p(\mathbf{A})\mathbf{b}$. This unified representation leads naturally to a common integral error representation and restarting mechanism.

Within this framework, we develop two classes of restarted algorithms. The first uses a fixed Krylov subspace dimension and is based either on the existing sketched Arnoldi process or on a new sketched harmonic Arnoldi process proposed in this work. These processes lead, respectively, to restarted sketched Arnoldi and sketched harmonic Arnoldi methods. The new process generates a nonorthonormal Krylov basis satisfying the sketched harmonic orthogonality condition needed for the restarting framework. The second chooses the Krylov subspace dimension adaptively by running the truncated Arnoldi process until the condition number of the generated basis, estimated from its sketch, exceeds a prescribed threshold. A rank-1 update is then applied so that the resulting Arnoldi-like decomposition satisfies the orthogonality condition required by the framework. We also provide a convergence analysis of the restarted harmonic Arnoldi and sketched harmonic Arnoldi methods.

The remainder of this paper is organized as follows. Section 2 formulates Krylov and sketched Krylov approximations within a common Arnoldi-like framework. Section 3 extends quadrature-based restarting to this framework. Section 4 presents the proposed algorithms, while Section 5 analyzes their convergence. Numerical experiments are presented in Section 6. Finally, Section 7 concludes the paper.

We use the following notation. The open right half-plane is denoted by $\mathbb{C}_+ = \{z \in \mathbb{C} : \Re(z) > 0\}$. The symbol $\|\cdot\|$ denotes the Euclidean norm for vectors and the spectral norm for matrices. For a matrix $\mathbf{A}$, we denote its spectrum by $\Lambda(\mathbf{A})$, its condition number by $\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$, and its numerical range by

$$\mathcal{W}(\mathbf{A}) := \{\mathbf{x}^* \mathbf{A} \mathbf{x} : \mathbf{x} \in \mathbb{C}^N, \|\mathbf{x}\| = 1\}.$$

A matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$ is called positive real if $\Re(\mathbf{x}^* \mathbf{A} \mathbf{x}) > 0$ for every nonzero $\mathbf{x} \in \mathbb{C}^N$. A matrix $\mathbf{V} \in \mathbb{C}^{N \times m}$ is called orthonormal if its columns form an orthonormal set. For a vector $\mathbf{x}$ and a subspace $\mathcal{V}$, we write $\mathbf{x} \perp \mathcal{V}$ if $\mathbf{x}$ is orthogonal to $\mathcal{V}$. If $\mathbf{V}$ is a basis of $\mathcal{V}$, this condition is equivalent to $\mathbf{V}^* \mathbf{x} = \mathbf{0}$, and we also write $\mathbf{x} \perp \mathbf{V}$. The set of polynomials of degree at most m is denoted by $\mathcal{P}_m$.

2. Krylov subspace methods and their sketched variants. This section places several Krylov and sketched Krylov approximations within a common Arnoldi-like framework, which serves as the algebraic foundation of the sketch-and-restart framework developed in the subsequent sections. Specifically, we show that the Arnoldi and harmonic Arnoldi approximations, together with their sketched variants, can all be expressed in the form $\mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta$, where $\mathbf{B}_m$ is a possibly nonorthonormal basis of a Krylov subspace and $\mathbf{H}_m$ is an upper Hessenberg matrix arising from an Arnoldi-like decomposition. This observation forms the algebraic foundation for the quadrature-based restarting mechanism developed in Section 3.

Suppose that f is analytic on the closure of a domain $\Omega \subset \mathbb{C}$ containing $\Lambda(\mathbf{A})$,

and admits the integral representation

$$f(z) = \int_{\Gamma} \frac{g(t)}{t-z} dt, \quad z \in \Omega, \quad (2.1)$$

where $\Gamma \subset \mathbb{C} \setminus \Omega$ is a contour and g is a suitable function on Γ . Such a representation follows, for example, from the Cauchy integral formula, in which case Γ encloses $\Lambda(\mathbf{A})$, and also includes Stieltjes functions such as $f(z) = z^{-\alpha}$ with $\alpha > 0$; see also [11]. Then

$$f(\mathbf{A})\mathbf{b} = \int_{\Gamma} g(t)(t\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} dt = \int_{\Gamma} g(t)\mathbf{x}(t) dt, \quad (2.2)$$

where $\mathbf{x}(t)$ solves the shifted linear system

$$(t\mathbf{I} - \mathbf{A})\mathbf{x}(t) = \mathbf{b}. \quad (2.3)$$

Let $\mathcal{K}_m(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{m-1}\mathbf{b}\}$ be the Krylov subspace generated by $\mathbf{A}$ and $\mathbf{b}$. Throughout this paper, we assume that the Krylov subspaces are nondegenerate, i.e., $\dim \mathcal{K}_j(\mathbf{A}, \mathbf{b}) = j$ for $1 \leq j \leq m$. An associated *Arnoldi-like decomposition* is

$$\begin{aligned} \mathbf{A}\mathbf{B}_m &= \mathbf{B}_m\mathbf{H}_m + \mathbf{b}_{m+1}h_{m+1,m}\mathbf{e}_m^\top \\ &= [\mathbf{B}_m \ \mathbf{b}_{m+1}] \begin{bmatrix} \mathbf{H}_m \\ h_{m+1,m}\mathbf{e}_m^\top \end{bmatrix} =: \mathbf{B}_{m+1}\underline{\mathbf{H}}_m, \end{aligned} \quad (2.4)$$

where $\mathbf{B}_m = [\mathbf{b}_1 \ \dots \ \mathbf{b}_m] \in \mathbb{C}^{N \times m}$ is a basis of $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$, and $\mathbf{H}_m \in \mathbb{C}^{m \times m}$ is upper Hessenberg. We assume that $\mathbf{b} = \mathbf{b}_1\beta$ for some $\beta \in \mathbb{C}$. When $\mathbf{B}_{m+1}$ is orthonormal, (2.4) reduces to the standard *Arnoldi decomposition*, which can be computed by the Arnoldi process [20].

2.1. Krylov subspace methods. In Krylov subspace methods, the solution $\mathbf{x}(t)$ of the shifted linear system (2.3) is approximated in $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ by $\mathbf{x}_m(t) = \mathbf{B}_m\mathbf{y}_m(t)$, where $\mathbf{y}_m(t) \in \mathbb{C}^m$. The corresponding residual is $\mathbf{r}_m(t) = \mathbf{b} - (t\mathbf{I} - \mathbf{A})\mathbf{x}_m(t)$. The coefficient vector $\mathbf{y}_m(t)$ is determined by imposing a *Petrov-Galerkin condition* on $\mathbf{r}_m(t)$.

If the projection condition is $\mathbf{r}_m(t) \perp \mathcal{K}_m(\mathbf{A}, \mathbf{b})$, then $\mathbf{x}_m(t)$ is the Arnoldi approximation to the solution of the shifted linear system (2.3). Solving the resulting reduced system gives

$$\mathbf{y}_m(t) = (t\mathbf{I}_m - \mathbf{B}_m^\dagger \mathbf{A} \mathbf{B}_m)^{-1} \mathbf{e}_1 \beta.$$

Replacing $\mathbf{x}(t)$ in (2.2) by $\mathbf{x}_m(t)$, we obtain

$$\begin{aligned} \mathbf{f}_m &= \int_{\Gamma} g(t)\mathbf{x}_m(t) dt = \mathbf{B}_m \left(\int_{\Gamma} g(t)(t\mathbf{I}_m - \mathbf{B}_m^\dagger \mathbf{A} \mathbf{B}_m)^{-1} dt \right) \mathbf{e}_1 \beta \\ &= \mathbf{B}_m f(\mathbf{B}_m^\dagger \mathbf{A} \mathbf{B}_m) \mathbf{e}_1 \beta, \end{aligned} \quad (2.5)$$

provided that the corresponding matrix function is well-defined. The vector $\mathbf{f}_m$ in (2.5) is called the *Arnoldi approximation* to $f(\mathbf{A})\mathbf{b}$. If $\mathbf{b}_{m+1} \perp \mathbf{B}_m$ in the Arnoldi-like decomposition (2.4), then

$$\mathbf{B}_m^\dagger \mathbf{A} \mathbf{B}_m = \mathbf{H}_m \quad \text{and} \quad \mathbf{f}_m = \mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta. \quad (2.6)$$

Thus, under this orthogonality condition, the Arnoldi approximation can be computed directly from the upper Hessenberg matrix $\mathbf{H}_m$. This condition holds, in particular, when $\mathbf{B}_{m+1}$ is generated by the standard Arnoldi process.

If the projection condition is $\mathbf{r}_m(t) \perp \mathcal{AK}_m(\mathbf{A}, \mathbf{b})$, then $\mathbf{x}_m(t)$ is the harmonic Arnoldi approximation to the solution of the shifted linear system (2.3). By a similar argument,

$$\mathbf{y}_m(t) = \left(t\mathbf{I}_m - ((\mathbf{AB}_m)^* \mathbf{B}_m)^{-1} (\mathbf{AB}_m)^* (\mathbf{AB}_m) \right)^{-1} \mathbf{e}_1 \beta.$$

The corresponding *harmonic Arnoldi* (*HmArnoldi*) approximation to $f(\mathbf{A})\mathbf{b}$ is

$$\mathbf{f}_m = \mathbf{B}_m f \left(((\mathbf{AB}_m)^* \mathbf{B}_m)^{-1} (\mathbf{AB}_m)^* (\mathbf{AB}_m) \right) \mathbf{e}_1 \beta, \quad (2.7)$$

provided that the corresponding matrix function is well-defined. If $\mathbf{b}_{m+1} \perp \mathbf{AB}_m$ in the Arnoldi-like decomposition (2.4), then

$$((\mathbf{AB}_m)^* \mathbf{B}_m)^{-1} (\mathbf{AB}_m)^* (\mathbf{AB}_m) = \mathbf{H}_m \quad \text{and} \quad \mathbf{f}_m = \mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta. \quad (2.8)$$

Thus, under this orthogonality condition, the harmonic Arnoldi approximation has the same form as the Arnoldi approximation in (2.6).

2.2. Sketched Krylov subspace methods. In the Krylov subspace methods introduced in Section 2.1, the shifted linear system (2.3) is approximated by imposing a Petrov–Galerkin condition on the residual. This procedure can be accelerated using the so-called *sketch-and-solve* approach [21]. More specifically, suppose that, for a fixed constant $\varepsilon \in [0, 1)$, the matrix $\mathbf{S} \in \mathbb{C}^{s \times N}$ satisfies the following *subspace embedding property* for all $\mathbf{v} \in \mathcal{K}_{m+1}(\mathbf{A}, \mathbf{b})$:

$$(1 - \varepsilon) \|\mathbf{v}\|^2 \leq \|\mathbf{Sv}\|^2 \leq (1 + \varepsilon) \|\mathbf{v}\|^2. \quad (2.9)$$

Condition (2.9) can equivalently be expressed in terms of inner products: for all $\mathbf{u}, \mathbf{v} \in \mathcal{K}_{m+1}(\mathbf{A}, \mathbf{b})$,

$$|\langle \mathbf{u}, \mathbf{v} \rangle - \langle \mathbf{Su}, \mathbf{Sv} \rangle| \leq \varepsilon \|\mathbf{u}\| \|\mathbf{v}\|. \quad (2.10)$$

Indeed, let $\mathbf{V}_{m+1} \in \mathbb{C}^{N \times (m+1)}$ have orthonormal columns spanning $\mathcal{K}_{m+1}(\mathbf{A}, \mathbf{b})$, and define $\mathbf{E} = \mathbf{V}_{m+1}^* (\mathbf{I} - \mathbf{S}^* \mathbf{S}) \mathbf{V}_{m+1}$. The matrix $\mathbf{E}$ is Hermitian. For any $\mathbf{z} \in \mathbb{C}^{m+1}$, the vector $\mathbf{v} = \mathbf{V}_{m+1} \mathbf{z}$ belongs to $\mathcal{K}_{m+1}(\mathbf{A}, \mathbf{b})$, and (2.9) gives

$$|\mathbf{z}^* \mathbf{E} \mathbf{z}| = |\|\mathbf{v}\|^2 - \|\mathbf{Sv}\|^2| \leq \varepsilon \|\mathbf{v}\|^2 = \varepsilon \|\mathbf{z}\|^2.$$

Since $\mathbf{E}$ is Hermitian, it follows that $\|\mathbf{E}\| = \max_{\|\mathbf{z}\|=1} |\mathbf{z}^* \mathbf{E} \mathbf{z}| \leq \varepsilon$. Now, for any $\mathbf{u}, \mathbf{v} \in \mathcal{K}_{m+1}(\mathbf{A}, \mathbf{b})$, write $\mathbf{u} = \mathbf{V}_{m+1} \mathbf{x}$ and $\mathbf{v} = \mathbf{V}_{m+1} \mathbf{y}$. Then

$$|\langle \mathbf{u}, \mathbf{v} \rangle - \langle \mathbf{Su}, \mathbf{Sv} \rangle| = |\mathbf{x}^* \mathbf{E} \mathbf{y}| \leq \|\mathbf{E}\| \|\mathbf{x}\| \|\mathbf{y}\| \leq \varepsilon \|\mathbf{u}\| \|\mathbf{v}\|,$$

which proves (2.10). Conversely, taking $\mathbf{u} = \mathbf{v}$ in (2.10) yields (2.9).

We call such a matrix $\mathbf{S}$ a *sketching matrix*, and s the sketching dimension. Two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{C}^N$ are called $\mathbf{S}$ -orthogonal, denoted by $\mathbf{x} \perp_{\mathbf{S}} \mathbf{y}$, if $\mathbf{Sx} \perp \mathbf{Sy}$. When $\mathbf{S}$ is chosen randomly [15], for example, as a Gaussian or sparse random matrix, the subspace embedding property (2.9) holds with high probability, provided that s is a small multiple of m , such as $s = 2m$.

We first consider the sketched version of the Arnoldi approximation. Let $\hat{\mathbf{x}}_m(t) = \mathbf{B}_m \hat{\mathbf{y}}_m(t)$ approximate the solution $\mathbf{x}(t)$ of (2.3), and define the corresponding residual by $\hat{\mathbf{r}}_m(t) = \mathbf{b} - (t\mathbf{I} - \mathbf{A})\hat{\mathbf{x}}_m(t)$. The Petrov–Galerkin condition is imposed after sketching, namely, $\hat{\mathbf{r}}_m(t) \perp_{\mathcal{S}} \mathcal{K}_m(\mathbf{A}, \mathbf{b})$. By the subspace embedding property (2.9), $\mathbf{S}\mathbf{B}_m$ has full column rank. Following the same argument as in Section 2.1, we obtain

$$\hat{\mathbf{y}}_m(t) = (t\mathbf{I}_m - (\mathbf{S}\mathbf{B}_m)^\dagger(\mathbf{S}\mathbf{A}\mathbf{B}_m))^{-1} \mathbf{e}_1 \beta$$

and

$$\hat{\mathbf{f}}_m = \mathbf{B}_m f((\mathbf{S}\mathbf{B}_m)^\dagger(\mathbf{S}\mathbf{A}\mathbf{B}_m)) \mathbf{e}_1 \beta, \quad (2.11)$$

provided that the corresponding matrix function is well-defined. The vector $\hat{\mathbf{f}}_m$ in (2.11) is called the *sketched Arnoldi (sArnoldi) approximation* to $f(\mathbf{A})\mathbf{b}$. We point out that the sArnoldi approximation (2.11) used here differs in form from that in [13], although the two formulations are mathematically equivalent. Moreover, if $\mathbf{b}_{m+1} \perp_{\mathcal{S}} \mathbf{B}_m$ in the Arnoldi-like decomposition (2.4), then

$$(\mathbf{S}\mathbf{B}_m)^\dagger(\mathbf{S}\mathbf{A}\mathbf{B}_m) = \mathbf{H}_m \quad \text{and} \quad \hat{\mathbf{f}}_m = \mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta. \quad (2.12)$$

A similar discussion applies to the HmArnoldi approximation. By imposing the condition $\hat{\mathbf{r}}_m(t) \perp_{\mathcal{S}} \mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{b})$, we obtain

$$\hat{\mathbf{y}}_m(t) = \left(t\mathbf{I}_m - ((\mathbf{S}\mathbf{A}\mathbf{B}_m)^*(\mathbf{S}\mathbf{B}_m))^{-1}(\mathbf{S}\mathbf{A}\mathbf{B}_m)^*(\mathbf{S}\mathbf{A}\mathbf{B}_m) \right)^{-1} \mathbf{e}_1 \beta.$$

Consequently, $\hat{\mathbf{x}}_m(t) = \mathbf{B}_m \hat{\mathbf{y}}_m(t)$ is the sketched harmonic Arnoldi approximation to the solution of the shifted linear system (2.3). The corresponding *sketched harmonic Arnoldi (sHmArnoldi) approximation* to $f(\mathbf{A})\mathbf{b}$ is

$$\hat{\mathbf{f}}_m = \mathbf{B}_m f\left(((\mathbf{S}\mathbf{A}\mathbf{B}_m)^*(\mathbf{S}\mathbf{B}_m))^{-1}(\mathbf{S}\mathbf{A}\mathbf{B}_m)^*(\mathbf{S}\mathbf{A}\mathbf{B}_m) \right) \mathbf{e}_1 \beta, \quad (2.13)$$

provided that the corresponding matrix function is well-defined. If $\mathbf{b}_{m+1} \perp_{\mathcal{S}} \mathbf{A}\mathbf{B}_m$ in the Arnoldi-like decomposition (2.4), then

$$((\mathbf{S}\mathbf{A}\mathbf{B}_m)^*(\mathbf{S}\mathbf{B}_m))^{-1}(\mathbf{S}\mathbf{A}\mathbf{B}_m)^*(\mathbf{S}\mathbf{A}\mathbf{B}_m) = \mathbf{H}_m \quad \text{and} \quad \hat{\mathbf{f}}_m = \mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta. \quad (2.14)$$

2.3. Rank-1 update of the Arnoldi-like decomposition. The discussions in Sections 2.1 and 2.2 show that enforcing a suitable orthogonality condition on the last vector $\mathbf{b}_{m+1}$ can simplify the approximation formula by reducing it to a matrix function of the upper Hessenberg matrix $\mathbf{H}_m$. This condition is not always satisfied by a given Arnoldi-like decomposition. It can, however, be enforced by applying a rank-1 update to (2.4), as described below; see also [5, 12] for related approaches.

Given (2.4), we update $\mathbf{b}_{m+1}$ to $\tilde{\mathbf{b}}_{m+1}$ so that $\tilde{\mathbf{b}}_{m+1}$ satisfies a prescribed orthogonality condition. To this end, write $\mathbf{b}_{m+1} = \mathbf{B}_m \mathbf{c}_m + \tilde{\mathbf{b}}_{m+1} \alpha_{m+1}$, where $\mathbf{c}_m$ is to be determined and α_{m+1} is chosen according to a normalization condition. Substituting this expression into (2.4) gives

$$\begin{aligned} \mathbf{A}\mathbf{B}_m &= \mathbf{B}_m \mathbf{H}_m + (\mathbf{B}_m \mathbf{c}_m + \tilde{\mathbf{b}}_{m+1} \alpha_{m+1}) h_{m+1,m} \mathbf{e}_m^\top \\ &= \mathbf{B}_m (\mathbf{H}_m + \mathbf{c}_m h_{m+1,m} \mathbf{e}_m^\top) + \tilde{\mathbf{b}}_{m+1} (\alpha_{m+1} h_{m+1,m}) \mathbf{e}_m^\top \\ &= \mathbf{B}_m \tilde{\mathbf{H}}_m + \tilde{\mathbf{b}}_{m+1} \tilde{h}_{m+1,m} \mathbf{e}_m^\top, \end{aligned}$$

where $\widetilde{\mathbf{H}}_m = \mathbf{H}_m + \mathbf{c}_m h_{m+1,m} \mathbf{e}_m^\top$ and $\tilde{h}_{m+1,m} = \alpha_{m+1} h_{m+1,m}$. The matrix $\widetilde{\mathbf{H}}_m$ remains upper Hessenberg.

If the desired orthogonality condition is $\tilde{\mathbf{b}}_{m+1} \perp \mathbf{B}_m$ or $\tilde{\mathbf{b}}_{m+1} \perp_{\mathcal{S}} \mathbf{B}_m$, then $\mathbf{c}_m = \mathbf{B}_m^\dagger \mathbf{b}_{m+1}$ or $\mathbf{c}_m = (\mathbf{S}\mathbf{B}_m)^\dagger (\mathbf{S}\mathbf{b}_{m+1})$, respectively. Similarly, if the desired orthogonality condition is $\tilde{\mathbf{b}}_{m+1} \perp \mathbf{A}\mathbf{B}_m$ or $\tilde{\mathbf{b}}_{m+1} \perp_{\mathcal{S}} \mathbf{A}\mathbf{B}_m$, then

$$\begin{aligned} \mathbf{c}_m &= ((\mathbf{A}\mathbf{B}_m)^* \mathbf{B}_m)^{-1} (\mathbf{A}\mathbf{B}_m)^* \mathbf{b}_{m+1} \\ \text{or } \mathbf{c}_m &= ((\mathbf{S}\mathbf{A}\mathbf{B}_m)^* (\mathbf{S}\mathbf{B}_m))^{-1} (\mathbf{S}\mathbf{A}\mathbf{B}_m)^* (\mathbf{S}\mathbf{b}_{m+1}), \end{aligned}$$

respectively.

3. Restarting based on quadrature. In this section, we extend quadrature-based restarting to the Arnoldi-like framework introduced in Section 2. Our extension builds on the restarting mechanism of [11]. This extension is central to the sketch-and-restart framework: any approximation of the form $\mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta$ admits an integral error representation, which can then be used for restarting. The key ingredient is the integral representation of the interpolating polynomial, stated in the following lemma [11, Lemma 3.1].

LEMMA 3.1 (Integral representation of the interpolating polynomial). *Let $\Omega \subset \mathbb{C}$ be a domain, and let $f: \Omega \rightarrow \mathbb{C}$ be analytic on the closure of Ω and admit the integral representation (2.1). Then the polynomial $p_{m-1} \in \mathcal{P}_{m-1}$ interpolating f at the points $\{\theta_1, \dots, \theta_m\} \subset \Omega$ is given by*

$$p_{m-1}(z) = \int_{\Gamma} \left(1 - \frac{\phi_m(z)}{\phi_m(t)}\right) \frac{g(t)}{t - z} dt, \quad (3.1)$$

where $\phi_m(z) = (z - \theta_1) \cdots (z - \theta_m)$, provided that the integral in (3.1) exists.

3.1. Restarting the Arnoldi-like approximation. We first recall the following polynomial identity for Arnoldi-like decompositions [19, Lemma 2.1].

LEMMA 3.2 (Polynomial action in an Arnoldi-like decomposition). *Given an Arnoldi-like decomposition (2.4), let $p(z) = c_0 + c_1 z + \cdots + c_m z^m \in \mathcal{P}_m$. Then*

$$p(\mathbf{A})\mathbf{b}_1 = \mathbf{B}_m p(\mathbf{H}_m) \mathbf{e}_1 + \mathbf{b}_{m+1} \gamma_m c_m,$$

where $\gamma_m = \prod_{j=1}^m h_{j+1,j}$.

The Arnoldi-like approximation to $f(\mathbf{A})\mathbf{b}$ is defined by

$$\tilde{\mathbf{f}}_m := \mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta. \quad (3.2)$$

Let $p_{m-1} \in \mathcal{P}_{m-1}$ interpolate f at $\Lambda(\mathbf{H}_m)$. By Lemma 3.2,

$$\tilde{\mathbf{f}}_m = \mathbf{B}_m p_{m-1}(\mathbf{H}_m) \mathbf{e}_1 \beta = p_{m-1}(\mathbf{A})\mathbf{b}_1 \beta = p_{m-1}(\mathbf{A})\mathbf{b}.$$

Suppose that $\Lambda(\mathbf{H}_m) = \{\theta_1, \dots, \theta_m\}$, and define $\phi_m(z) = (z - \theta_1) \cdots (z - \theta_m)$. The eigenvalues are counted with algebraic multiplicity; in the case of multiple eigenvalues, the interpolation is understood in the Hermite sense. Then Lemma 3.1 gives

$$f(z) - p_{m-1}(z) = \int_{\Gamma} \frac{\phi_m(z)}{\phi_m(t)} \frac{g(t)}{t - z} dt,$$

and therefore

$$f(\mathbf{A})\mathbf{b} - \tilde{\mathbf{f}}_m = \left(\int_{\Gamma} \frac{g(t)}{\phi_m(t)} (t\mathbf{I} - \mathbf{A})^{-1} dt \right) \phi_m(\mathbf{A})\mathbf{b}.$$

Moreover, Lemma 3.2 gives

$$\phi_m(\mathbf{A})\mathbf{b} = \mathbf{B}_m\phi_m(\mathbf{H}_m)\mathbf{e}_1\beta + \mathbf{b}_{m+1}\gamma_m\beta = \mathbf{b}_{m+1}\gamma_m\beta,$$

where we used $\phi_m(\mathbf{H}_m) = \mathbf{0}$. Thus,

$$f(\mathbf{A})\mathbf{b} - \tilde{\mathbf{f}}_m = \gamma_m\beta \left(\int_{\Gamma} \frac{g(t)}{\phi_m(t)} (t\mathbf{I} - \mathbf{A})^{-1} dt \right) \mathbf{b}_{m+1}.$$

The following theorem generalizes Theorem 3.4 of [11].

THEOREM 3.3 (Error of the Arnoldi-like approximation). *Consider the Arnoldi-like decomposition (2.4) with $\mathbf{b} = \mathbf{b}_1\beta$. Suppose that f admits the integral representation (2.1), and let $\tilde{\mathbf{f}}_m$ be the Arnoldi-like approximation defined in (3.2). Define $\phi_m(t) = (t - \theta_1) \cdots (t - \theta_m)$, where $\Lambda(\mathbf{H}_m) = \{\theta_1, \dots, \theta_m\} \subset \Omega$, and let $\gamma_m = \prod_{j=1}^m h_{j+1,j}$. Then*

$$f(\mathbf{A})\mathbf{b} - \tilde{\mathbf{f}}_m = \gamma_m\beta \left(\int_{\Gamma} \frac{g(t)}{\phi_m(t)} (t\mathbf{I} - \mathbf{A})^{-1} dt \right) \mathbf{b}_{m+1} =: \text{err}(\mathbf{A})\mathbf{b}_{m+1}, \quad (3.3)$$

provided that the integral exists. Thus, the error function is

$$\text{err}(z) = \gamma_m\beta \int_{\Gamma} \frac{g(t)}{\phi_m(t)} (t - z)^{-1} dt.$$

Using $\mathbf{b}_{m+1}$ as the new starting vector, suppose that a new Arnoldi-like decomposition $\mathbf{A}\mathbf{B}_m^+ = \mathbf{B}_{m+1}^+ \mathbf{H}_m^+$ is generated with $\mathbf{b}_{m+1} = \mathbf{b}_1^+ \beta^+$. Then the error term can itself be approximated by an Arnoldi-like approximation:

$$\text{err}(\mathbf{A})\mathbf{b}_{m+1} = \mathbf{B}_m^+ \text{err}(\mathbf{H}_m^+) \mathbf{e}_1 \beta^+ + \text{err}^+(\mathbf{A})\mathbf{b}_{m+1}^+,$$

where $\text{err}^+(z)$ is the new error function. Repeating this process yields the general restarting framework in Algorithm 1. The corresponding error functions $\text{err}^{[k]}(z)$ are given in Corollary 3.4. Although the corollary assumes a fixed Krylov subspace dimension m in each restart cycle, the result extends directly to variable dimensions.

Algorithm 1 Restarting framework for computing $f(\mathbf{A})\mathbf{b}$

Input: Matrix $\mathbf{A}$, vector $\mathbf{b}$, function f , subspace dimension m

Output: $\mathbf{f}^{[k]} \approx f(\mathbf{A})\mathbf{b}$

- 1: Compute an Arnoldi-like decomposition $\mathbf{A}\mathbf{B}_m^{[0]} = \mathbf{B}_{m+1}^{[0]} \mathbf{H}_m^{[0]}$ with $\mathbf{b} = \mathbf{b}_1^{[0]} \beta^{[0]}$.
 - 2: $\mathbf{f}^{[0]} = \mathbf{B}_m^{[0]} f(\mathbf{H}_m^{[0]}) \mathbf{e}_1 \beta^{[0]}$.
 - 3: **for** $k = 1, 2, \dots$ **do**
 - 4: Determine the error function $\text{err}^{[k]}(z)$.
 - 5: Compute an Arnoldi-like decomposition $\mathbf{A}\mathbf{B}_m^{[k]} = \mathbf{B}_{m+1}^{[k]} \mathbf{H}_m^{[k]}$ with $\mathbf{b}_{m+1}^{[k-1]} = \mathbf{b}_1^{[k]} \beta^{[k]}$.
 - 6: Compute $\mathbf{g}^{[k]} \approx \mathbf{B}_m^{[k]} \text{err}^{[k]}(\mathbf{H}_m^{[k]}) \mathbf{e}_1 \beta^{[k]}$.
 - 7: Update $\mathbf{f}^{[k]} = \mathbf{f}^{[k-1]} + \mathbf{g}^{[k]}$.
 - 8: **end for**
-

COROLLARY 3.4 (Error of the Arnoldi-like approximation after k restarts). *Under the assumptions of Theorem 3.3, suppose that the ℓ -th cycle generates an Arnoldi-like*

decomposition $\mathbf{A}\mathbf{B}_m^{[\ell]} = \mathbf{B}_{m+1}^{[\ell]}\mathbf{H}_m^{[\ell]}$ for $0 \leq \ell \leq k$. Here, $\mathbf{b} = \mathbf{b}_1^{[0]}\beta^{[0]}$ and $\mathbf{b}_{m+1}^{[\ell-1]} = \mathbf{b}_1^{[\ell]}\beta^{[\ell]}$ for $1 \leq \ell \leq k$. Assume that $\Lambda(\mathbf{H}_m^{[\ell]}) = \{\theta_j^{[\ell]}\}_{j=1}^m \subset \Omega$, and define $\phi_m^{[\ell]}(t) = \prod_{j=1}^m (t - \theta_j^{[\ell]})$ and $\gamma_m^{[\ell]} = \prod_{j=1}^m h_{j+1,j}^{[\ell]}$ for $0 \leq \ell \leq k$. Then the error after k restarts is

$$\begin{aligned} f(\mathbf{A})\mathbf{b} - \mathbf{f}^{[k]} &= \left(\prod_{\ell=0}^k \gamma_m^{[\ell]} \beta^{[\ell]} \right) \left(\int_{\Gamma} \frac{g(t)}{\prod_{\ell=0}^k \phi_m^{[\ell]}(t)} (t\mathbf{I} - \mathbf{A})^{-1} dt \right) \mathbf{b}_{m+1}^{[k]} \\ &=: \text{err}^{[k+1]}(\mathbf{A})\mathbf{b}_{m+1}^{[k]}. \end{aligned}$$

In practice, the error functions are evaluated using adaptive quadrature. For a given error function, we compute two quadrature approximations using r_1 and r_2 nodes, where $r_2 = 2r_1$. The approximation based on r_2 nodes is accepted if the corresponding reduced vectors differ by less than a prescribed tolerance. Otherwise, we set $r_1 = r_2$, double r_2 , and repeat the test until the tolerance is satisfied or a prescribed maximum number of nodes is reached. If no increase in the number of quadrature nodes is required in a given iteration, we reduce the quadrature sizes for the next iteration by setting $r_2 = r_1$ and halving r_1 .

3.2. Restarting Krylov and sketched Krylov subspace methods. The discussion above links the Arnoldi-like framework of Section 2 with the quadrature-based restarting mechanism. Indeed, the Arnoldi approximation (2.6), the HmArnoldi approximation (2.8), the sArnoldi approximation (2.12), and the sHmArnoldi approximation (2.14) are all special cases of the Arnoldi-like approximation (3.2), provided that the corresponding orthogonality conditions are satisfied. Thus, the same error representation and restarting mechanism apply to all these approximations. This observation forms the basis of the sketch-and-restart framework developed in the next section. Different basis generation processes lead to different restarted algorithms, while the underlying restarting formula remains unchanged.

4. Sketch-and-restart algorithms for matrix functions. The previous section showed that quadrature-based restarting applies to any Arnoldi-like approximation of the form $\mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta$, provided that the required orthogonality condition is satisfied. In this section, we instantiate this restarting mechanism using different strategies for generating Arnoldi-like decompositions. Specifically, we consider three such strategies: the existing sketched Arnoldi process, a new sketched harmonic Arnoldi process introduced in this work, and an adaptive truncated Arnoldi process combined with a rank-1 update. These strategies lead to two restarted algorithms for computing $f(\mathbf{A})\mathbf{b}$.

4.1. Basis generation. The standard Arnoldi process [20] generates an orthonormal basis for the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$, but its orthogonalization and storage costs can become substantial when m is large. In the sketch-and-restart framework, full orthogonality is not required. It suffices to construct an Arnoldi-like decomposition that satisfies the orthogonality condition required by the corresponding approximation. We therefore consider three basis generation strategies that produce possibly nonorthonormal bases with different computational and storage trade-offs.

The first strategy is the *sketched Arnoldi (s-Arnoldi) process* [2, 7]. It is based on the Arnoldi process but computes the Gram–Schmidt coefficients in the sketched space. The same coefficients are then used to update the full-space basis vectors, as summarized in Algorithm 2. For simplicity, we assume that no breakdown occurs. The process yields a standard Arnoldi decomposition in the sketched space. Consequently,

the resulting Arnoldi-like decomposition satisfies $\mathbf{b}_{m+1} \perp_{\mathbf{S}} \mathbf{B}_m$, and the sArnoldi approximation can be computed directly from (2.12). The complexity of the s-Arnoldi process is $\mathcal{O}(mT_{\text{apply}} + mT_{\text{sketching}} + Nm^2 + m^3)$, where T_{apply} and $T_{\text{sketching}}$ denote the costs of applying $\mathbf{A}$ and $\mathbf{S}$ to a vector, respectively. Compared with the standard Arnoldi process, s-Arnoldi reduces the orthogonalization cost by nearly one half, while the additional costs are of lower order.

Algorithm 2 Sketched Arnoldi (s-Arnoldi) process

Input: Matrix $\mathbf{A}$, vector $\mathbf{b}$, subspace dimension m , sketching matrix $\mathbf{S}$

Output: Arnoldi-like decomposition $\mathbf{AB}_m = \mathbf{B}_{m+1}\underline{\mathbf{H}}_m$ with $\mathbf{b} = \mathbf{b}_1\beta$

```

1:  $\tilde{\mathbf{b}}_1 = \mathbf{b}$ ,  $\tilde{\mathbf{p}}_1 = \mathbf{S}\tilde{\mathbf{b}}_1$ .
2:  $\beta = \|\tilde{\mathbf{p}}_1\|$ ,  $\mathbf{p}_1 = \tilde{\mathbf{p}}_1/\beta$ ,  $\mathbf{b}_1 = \tilde{\mathbf{b}}_1/\beta$ .
3: for  $j = 1, 2, \dots, m$  do
4:    $\tilde{\mathbf{b}}_{j+1} = \mathbf{A}\tilde{\mathbf{b}}_j$ ,  $\tilde{\mathbf{p}}_{j+1} = \mathbf{S}\tilde{\mathbf{b}}_{j+1}$ .
5:   for  $i = 1, \dots, j$  do
6:      $h_{i,j} = \langle \tilde{\mathbf{p}}_{j+1}, \mathbf{p}_i \rangle$ ,  $\tilde{\mathbf{p}}_{j+1} = \tilde{\mathbf{p}}_{j+1} - \mathbf{p}_i h_{i,j}$ ,  $\tilde{\mathbf{b}}_{j+1} = \tilde{\mathbf{b}}_{j+1} - \mathbf{b}_i h_{i,j}$ .
7:   end for
8:    $h_{j+1,j} = \|\tilde{\mathbf{p}}_{j+1}\|$ ,  $\mathbf{p}_{j+1} = \tilde{\mathbf{p}}_{j+1}/h_{j+1,j}$ ,  $\mathbf{b}_{j+1} = \tilde{\mathbf{b}}_{j+1}/h_{j+1,j}$ .
9: end for
```

The second strategy is the *sketched harmonic Arnoldi (sHm-Arnoldi) process*, which, to the best of our knowledge, is new. It is similar to the s-Arnoldi process but aims to generate a nonorthonormal basis such that $\mathbf{b}_{j+1} \perp_{\mathbf{S}} \mathbf{AB}_j$ for $1 \leq j \leq m$. In particular, the resulting Arnoldi-like decomposition satisfies $\mathbf{b}_{m+1} \perp_{\mathbf{S}} \mathbf{AB}_m$, so the sHmArnoldi approximation can be computed directly from (2.14). The procedure, summarized in Algorithm 3, orthogonalizes each new sketched basis vector against the sketched images of the current basis vectors under $\mathbf{A}$, thereby enforcing $\langle \mathbf{p}_{j+1}, \mathbf{z}_i \rangle = 0$ for $1 \leq i \leq j$, where $\mathbf{z}_i = \mathbf{q}_i/\rho_i^*$ is scaled so that $\langle \mathbf{p}_i, \mathbf{z}_i \rangle = 1$. The complexity of sHm-Arnoldi is $\mathcal{O}(mT_{\text{apply}} + mT_{\text{sketching}} + Nm^2 + m^3)$, and its orthogonalization cost is nearly one half of that of the standard Arnoldi process.

Algorithm 3 Sketched harmonic Arnoldi (sHm-Arnoldi) process

Input: Matrix $\mathbf{A}$, vector $\mathbf{b}$, subspace dimension m , sketching matrix $\mathbf{S}$

Output: Arnoldi-like decomposition $\mathbf{AB}_m = \mathbf{B}_{m+1}\underline{\mathbf{H}}_m$ with $\mathbf{b} = \mathbf{b}_1\beta$

```

1:  $\tilde{\mathbf{b}}_1 = \mathbf{b}$ ,  $\tilde{\mathbf{p}}_1 = \mathbf{S}\tilde{\mathbf{b}}_1$ .
2:  $\beta = \|\tilde{\mathbf{p}}_1\|$ ,  $\mathbf{p}_1 = \tilde{\mathbf{p}}_1/\beta$ ,  $\mathbf{b}_1 = \tilde{\mathbf{b}}_1/\beta$ .
3: for  $j = 1, 2, \dots, m$  do
4:    $\tilde{\mathbf{b}}_{j+1} = \mathbf{A}\tilde{\mathbf{b}}_j$ ,  $\tilde{\mathbf{p}}_{j+1} = \mathbf{S}\tilde{\mathbf{b}}_{j+1}$ ,  $\mathbf{q}_j = \tilde{\mathbf{p}}_{j+1}$ .
5:    $\rho_j = \langle \mathbf{p}_j, \mathbf{q}_j \rangle$ ,  $\mathbf{z}_j = \mathbf{q}_j/\rho_j^*$ .
6:   for  $i = 1, \dots, j$  do
7:      $h_{i,j} = \langle \tilde{\mathbf{p}}_{j+1}, \mathbf{z}_i \rangle$ ,  $\tilde{\mathbf{p}}_{j+1} = \tilde{\mathbf{p}}_{j+1} - \mathbf{p}_i h_{i,j}$ ,  $\tilde{\mathbf{b}}_{j+1} = \tilde{\mathbf{b}}_{j+1} - \mathbf{b}_i h_{i,j}$ .
8:   end for
9:    $h_{j+1,j} = \|\tilde{\mathbf{p}}_{j+1}\|$ .
10:   $\mathbf{p}_{j+1} = \tilde{\mathbf{p}}_{j+1}/h_{j+1,j}$ ,  $\mathbf{b}_{j+1} = \tilde{\mathbf{b}}_{j+1}/h_{j+1,j}$ .
11: end for
```

The third strategy is the *truncated Arnoldi (t-Arnoldi) process* [20]. It generates a nonorthonormal basis by orthogonalizing each new vector only against the t most

recently generated basis vectors. We use this process adaptively by monitoring the condition number of the sketched basis $\mathbf{S}\mathbf{B}_m$. Once this condition number exceeds a prescribed threshold τ , the process is terminated, and a rank-1 update is applied to enforce the required orthogonality condition, as discussed in Section 2.3. Since the resulting Krylov subspace dimension is typically small, the rank-1 update is inexpensive and can be performed at the BLAS-2 level. The complete procedure is summarized in Algorithm 4 and has complexity $\mathcal{O}(mT_{\text{apply}} + mT_{\text{sketching}} + Nm^2)$.

Algorithm 4 Truncated Arnoldi (t -Arnoldi) process

Input: Matrix $\mathbf{A}$, vector $\mathbf{b}$, truncation parameter t , sketching matrix $\mathbf{S}$, condition number threshold τ , maximum Krylov subspace dimension m_{max} , sketching dimension increment s_0 , sketching dimension control parameter η

Output: Arnoldi-like decomposition $\mathbf{A}\mathbf{B}_m = \mathbf{B}_{m+1}\mathbf{H}_m$ with $\mathbf{b} = \mathbf{b}_1\beta$

- 1: $\tilde{\mathbf{b}}_1 = \mathbf{b}$, $\beta = \|\tilde{\mathbf{b}}_1\|$, $\mathbf{b}_1 = \tilde{\mathbf{b}}_1/\beta$.
 - 2: **if** $\mathbf{S}$ is empty **then**
 - 3: Draw a sketching matrix $\mathbf{S} \in \mathbb{C}^{s_0 \times N}$.
 - 4: **end if**
 - 5: Let s be the number of rows of $\mathbf{S}$.
 - 6: $\mathbf{P}_1 = [\mathbf{S}\mathbf{b}_1]$.
 - 7: **for** $j = 1, \dots, m_{\text{max}}$ **do**
 - 8: $\tilde{\mathbf{b}}_{j+1} = \mathbf{A}\mathbf{b}_j$.
 - 9: **for** $i = \max(1, j - t + 1), \dots, j$ **do**
 - 10: $h_{i,j} = \langle \tilde{\mathbf{b}}_{j+1}, \mathbf{b}_i \rangle$, $\tilde{\mathbf{b}}_{j+1} = \tilde{\mathbf{b}}_{j+1} - \mathbf{b}_i h_{i,j}$.
 - 11: **end for**
 - 12: $h_{j+1,j} = \|\tilde{\mathbf{b}}_{j+1}\|$, $\mathbf{b}_{j+1} = \tilde{\mathbf{b}}_{j+1}/h_{j+1,j}$.
 - 13: $\mathbf{P}_{j+1} = [\mathbf{P}_j, \mathbf{S}\mathbf{b}_{j+1}]$.
 - 14: **if** $\kappa(\mathbf{P}_{j+1}) > \tau$ **then**
 - 15: Break.
 - 16: **end if**
 - 17: **if** $s < \eta(j + 1)$ **then**
 - 18: Draw a sketching matrix $\mathbf{S}_{\text{incr}} \in \mathbb{C}^{s_0 \times N}$.
 - 19: Update $\mathbf{S} \leftarrow \begin{bmatrix} \mathbf{S} \\ \mathbf{S}_{\text{incr}} \end{bmatrix}$ and $\mathbf{P}_{j+1} \leftarrow \begin{bmatrix} \mathbf{P}_{j+1} \\ \mathbf{S}_{\text{incr}}\mathbf{b}_{j+1} \end{bmatrix}$.
 - 20: $s \leftarrow s + s_0$.
 - 21: **end if**
 - 22: **end for**
 - 23: Set $m = j$.
 - 24: Perform a rank-1 update as described in Section 2.3 so that the required orthogonality condition is satisfied.
-

4.2. Algorithms. We now present two algorithmic realizations of the sketch-and-restart framework. Algorithm 5 uses a fixed Krylov subspace dimension and applies either the s-Arnoldi or the sHm-Arnoldi process in each restart cycle, leading respectively to the restarted sArnoldi and restarted sHmArnoldi methods. The error function is updated as in Algorithm 1, and the correction term is evaluated by quadrature. Algorithm 6 instead chooses the Krylov subspace dimension adaptively using the t -Arnoldi process. The dimension m_0 obtained in the first restart cycle is used as the upper bound m_{max} in all subsequent cycles. The same quadrature-based restarting mechanism is then applied to the resulting Arnoldi-like decompositions.

Algorithm 5 Restarted sketched Krylov subspace methods for computing $f(\mathbf{A})\mathbf{b}$ **Input:** Matrix $\mathbf{A}$, vector $\mathbf{b}$, function f , subspace dimension m , sketching dimension s **Output:** $\mathbf{f}^{[k]} \approx f(\mathbf{A})\mathbf{b}$

- 1: Generate a sketching matrix $\mathbf{S} \in \mathbb{C}^{s \times N}$.
- 2: Compute an Arnoldi-like decomposition $\mathbf{A}\mathbf{B}_m^{[0]} = \mathbf{B}_{m+1}^{[0]}\mathbf{H}_m^{[0]}$ with $\mathbf{b} = \mathbf{b}_1^{[0]}\beta^{[0]}$ using the s-Arnoldi process (Algorithm 2) or the sHm-Arnoldi process (Algorithm 3).
- 3: $\mathbf{f}^{[0]} = \mathbf{B}_m^{[0]}f(\mathbf{H}_m^{[0]})\mathbf{e}_1\beta^{[0]}$.
- 4: **for** $k = 1, 2, \dots$ until convergence **do**
- 5: Determine the error function $\text{err}^{[k]}(z)$.
- 6: Compute an Arnoldi-like decomposition $\mathbf{A}\mathbf{B}_m^{[k]} = \mathbf{B}_{m+1}^{[k]}\mathbf{H}_m^{[k]}$ with $\mathbf{b}_{m+1}^{[k-1]} = \mathbf{b}_1^{[k]}\beta^{[k]}$ using the s-Arnoldi process (Algorithm 2) or the sHm-Arnoldi process (Algorithm 3).
- 7: Compute $\mathbf{g}^{[k]} \approx \mathbf{B}_m^{[k]}\text{err}^{[k]}(\mathbf{H}_m^{[k]})\mathbf{e}_1\beta^{[k]}$ using quadrature.
- 8: Update $\mathbf{f}^{[k]} = \mathbf{f}^{[k-1]} + \mathbf{g}^{[k]}$.
- 9: **end for**

Algorithm 6 Restarted adaptive Krylov subspace methods for computing $f(\mathbf{A})\mathbf{b}$ **Input:** Matrix $\mathbf{A}$, vector $\mathbf{b}$, function f , truncation parameter t , condition number threshold τ , maximum Krylov subspace dimension $m_{\max}$, sketching dimension increment s_0 , sketching dimension control parameter η **Output:** $\mathbf{f}^{[k]} \approx f(\mathbf{A})\mathbf{b}$

- 1: Use Algorithm 4 to obtain an Arnoldi-like decomposition $\mathbf{A}\mathbf{B}_{m_0}^{[0]} = \mathbf{B}_{m_0+1}^{[0]}\mathbf{H}_{m_0}^{[0]}$ with $\mathbf{b} = \mathbf{b}_1^{[0]}\beta^{[0]}$, as well as the sketching matrix $\mathbf{S}$.
- 2: Set $m_{\max} = m_0$.
- 3: $\mathbf{f}^{[0]} = \mathbf{B}_{m_0}^{[0]}f(\mathbf{H}_{m_0}^{[0]})\mathbf{e}_1\beta^{[0]}$.
- 4: **for** $k = 1, 2, \dots$ until convergence **do**
- 5: Determine the error function $\text{err}^{[k]}(z)$.
- 6: Use Algorithm 4 to obtain an Arnoldi-like decomposition $\mathbf{A}\mathbf{B}_{m_k}^{[k]} = \mathbf{B}_{m_k+1}^{[k]}\mathbf{H}_{m_k}^{[k]}$ with $\mathbf{b}_{m_{k-1}+1}^{[k-1]} = \mathbf{b}_1^{[k]}\beta^{[k]}$.
- 7: Compute $\mathbf{g}^{[k]} \approx \mathbf{B}_{m_k}^{[k]}\text{err}^{[k]}(\mathbf{H}_{m_k}^{[k]})\mathbf{e}_1\beta^{[k]}$ using quadrature.
- 8: Update $\mathbf{f}^{[k]} = \mathbf{f}^{[k-1]} + \mathbf{g}^{[k]}$.
- 9: **end for**

5. Convergence analysis. In this section, we analyze the convergence of the proposed sketch-and-restart algorithms. We assume exact arithmetic and neglect quadrature errors, so the analysis focuses on the Krylov and sketching components of the methods. The non-sketched adaptive variants can be related directly to existing restarted Arnoldi and harmonic Arnoldi methods through the Arnoldi-like framework developed above. The main new issue is the convergence of the restarted sketched harmonic Arnoldi method, for which the Petrov–Galerkin condition is imposed only after applying the sketching matrix.

Our analysis extends the GMRES-based approach of [10] to the sketched setting. This approach is particularly useful because it does not rely on CG-type arguments and therefore remains applicable in the presence of complex Ritz values. We focus on

Stieltjes functions of the form

$$f(z) = \int_{-\infty}^0 \frac{g(t)}{t-z} dt, \quad (5.1)$$

where g is a suitable nonpositive function on $(-\infty, 0]$.

5.1. Harmonic Arnoldi approximation with a nonorthogonal basis. We first consider the non-sketched adaptive variants of Algorithm 6. We show that, after the rank-1 update, these variants are mathematically equivalent to the restarted Arnoldi and HmArnoldi methods analyzed in [10], even though the basis generated by the t -Arnoldi process is not orthonormal.

Assume that we have an Arnoldi-like decomposition (2.4), where $\mathbf{B}_m$ has full column rank but is not necessarily orthonormal. We further assume that $\mathbf{b}_{m+1} \perp \mathbf{A}\mathbf{B}_m$, which can be enforced by the rank-1 update discussed in Section 2.3. Under this assumption, Algorithm 6 is mathematically equivalent to the restarted HmArnoldi method analyzed in [10]. The corresponding HmArnoldi approximation to the solution of the shifted linear system (2.3) is $\mathbf{x}_m(t) = \mathbf{B}_m(t\mathbf{I}_m - \mathbf{H}_m)^{-1}\mathbf{e}_1\beta$. Its residual satisfies

$$\begin{aligned} \mathbf{r}_m(t) &= \mathbf{b} - (t\mathbf{I} - \mathbf{A})\mathbf{x}_m(t) \\ &= \mathbf{b} - (t\mathbf{I} - \mathbf{A})\mathbf{B}_m(t\mathbf{I}_m - \mathbf{H}_m)^{-1}\mathbf{e}_1\beta \\ &= \mathbf{b} - [\mathbf{B}_m \ \mathbf{b}_{m+1}](t\mathbf{I}_m - \mathbf{H}_m)(t\mathbf{I}_m - \mathbf{H}_m)^{-1}\mathbf{e}_1\beta \\ &= \mathbf{b}_{m+1}h_{m+1,m}\mathbf{e}_m^\top(t\mathbf{I}_m - \mathbf{H}_m)^{-1}\mathbf{e}_1\beta \\ &=: \mathbf{b}_{m+1}\zeta(t). \end{aligned}$$

Thus, the residual is collinear with $\mathbf{b}_{m+1}$ for all t .

For each t , let $p_{m-1,t} \in \mathcal{P}_{m-1}$ be the polynomial that interpolates $h_t(z) = (t-z)^{-1}$ at the harmonic Ritz values $\Lambda(\mathbf{H}_m)$. Then Lemma 3.2 gives

$$\mathbf{x}_m(t) = \mathbf{B}_m(t\mathbf{I}_m - \mathbf{H}_m)^{-1}\mathbf{e}_1\beta = p_{m-1,t}(\mathbf{A})\mathbf{b}.$$

If f is a Stieltjes function of the form (5.1), then the HmArnoldi approximation (2.8) can be written as $\mathbf{f}_m = \int_{-\infty}^0 g(t)\mathbf{x}_m(t) dt$. Up to the sign convention in the Stieltjes representation, this is precisely the class of HmArnoldi approximants analyzed in [10]. Consequently, the convergence results of [10] carry over to Algorithm 6. The following theorem summarizes this observation.

THEOREM 5.1 (Convergence of the adaptive restarted Arnoldi-like method). *Let f be a Stieltjes function, and consider Algorithm 6, where the basis is generated by the truncated Arnoldi process in Algorithm 4.*

If the rank-1 update enforces $\mathbf{b}_{m+1} \perp \mathbf{B}_m$, then the algorithm is mathematically equivalent to the restarted Arnoldi method for computing $f(\mathbf{A})\mathbf{b}$. Hence, if $\mathbf{A}$ is Hermitian positive definite, the method converges for every restart length $m \geq 1$.

If the rank-1 update enforces $\mathbf{b}_{m+1} \perp \mathbf{A}\mathbf{B}_m$, then the algorithm is mathematically equivalent to the restarted HmArnoldi method for computing $f(\mathbf{A})\mathbf{b}$. Hence, if $\mathbf{A}$ is positive real, the method converges for every restart length $m \geq 1$.

Proof. The equivalence to the restarted Arnoldi and restarted HmArnoldi methods follows from the Arnoldi-like framework and the rank-1 update described above. The convergence of the restarted Arnoldi method for Hermitian positive definite matrices is established in [10, Theorem 4.3], while the convergence of the restarted

HmArnoldi method for positive real matrices is proved in [10, Theorem 6.5]. Since both convergence results hold for every restart length $m \geq 1$, they remain valid when the restart length varies between restart cycles. $\square$

5.2. Sketched harmonic Arnoldi with a nonorthogonal basis. We now turn to the sketched setting, where the Petrov–Galerkin condition is imposed only after applying the sketching matrix. Unlike the non-sketched variants discussed above, the restarted sHmArnoldi method cannot be reduced directly to the HmArnoldi method analyzed in [10]. We therefore establish the residual relations needed to extend the GMRES-based convergence analysis to the sketched setting.

Suppose that, in the Arnoldi-like decomposition (2.4), the matrix $\mathbf{B}_m$ has full column rank but is not necessarily orthonormal. Assume further that $\mathbf{b}_{m+1} \perp_{\mathcal{S}} \mathbf{A}\mathbf{B}_m$. The sHmArnoldi approximation to the solution of the shifted linear system (2.3) is $\hat{\mathbf{x}}_m(t) = \mathbf{B}_m(t\mathbf{I}_m - \mathbf{H}_m)^{-1}\mathbf{e}_1\beta$. As in Section 5.1, its residual satisfies

$$\hat{\mathbf{r}}_m(t) = \mathbf{b} - (t\mathbf{I} - \mathbf{A})\hat{\mathbf{x}}_m(t) = \mathbf{b}_{m+1}\hat{\zeta}(t). \quad (5.2)$$

Thus, the residual is collinear with $\mathbf{b}_{m+1}$ for all t .

Similarly, for each t , there exists a polynomial $\hat{p}_{m-1,t} \in \mathcal{P}_{m-1}$ such that $\hat{\mathbf{x}}_m(t) = \hat{p}_{m-1,t}(\mathbf{A})\mathbf{b}$, where $\hat{p}_{m-1,t}$ interpolates $h_t(z) = (t-z)^{-1}$ at the sketched harmonic Ritz values $\Lambda(\mathbf{H}_m)$. Therefore, the residual can be written as

$$\hat{\mathbf{r}}_m(t) = \hat{q}_{m,t}(\mathbf{A})\mathbf{b}, \quad \hat{q}_{m,t}(z) = 1 - (t-z)\hat{p}_{m-1,t}(z).$$

The following lemma is the sketched analogue of [10, Lemma 6.2(iv)]. We include a direct proof for completeness.

LEMMA 5.2 (A formula for the residual). *Let $\Lambda(\mathbf{H}_m) = \{\theta_1, \dots, \theta_m\}$, and assume that $\theta_j \neq 0$ for $j = 1, \dots, m$. Then the residuals defined in (5.2) satisfy*

$$\hat{\mathbf{r}}_m(t) = \frac{1}{(1-t/\theta_1) \cdots (1-t/\theta_m)} \hat{\mathbf{r}}_m(0) =: \hat{\mathbf{r}}_m(0)\eta_m(t).$$

Proof. Let $\phi_m(z) = (z - \theta_1) \cdots (z - \theta_m)$. It is straightforward to verify that $\hat{p}_{m-1,t}(z) = (1 - \phi_m(z)/\phi_m(t))/(t-z)$ is the polynomial interpolating $h_t(z) = (t-z)^{-1}$ at $\Lambda(\mathbf{H}_m)$. Direct substitution gives $\hat{q}_{m,t}(z) = 1 - (t-z)\hat{p}_{m-1,t}(z) = \phi_m(z)/\phi_m(t)$. Therefore,

$$\hat{\mathbf{r}}_m(t) = \hat{q}_{m,t}(\mathbf{A})\mathbf{b} = \frac{\phi_m(\mathbf{A})}{\phi_m(t)}\mathbf{b} = \frac{\phi_m(0)}{\phi_m(t)} \frac{\phi_m(\mathbf{A})}{\phi_m(0)}\mathbf{b} = \frac{\phi_m(0)}{\phi_m(t)}\hat{\mathbf{r}}_m(0).$$

Since $\phi_m(t)/\phi_m(0) = (1-t/\theta_1) \cdots (1-t/\theta_m)$, the stated formula follows. $\square$

Suppose that $\mathbf{A}$ is positive real. If the sketched harmonic Ritz values $\Lambda(\mathbf{H}_m)$ lie in the open right half-plane $\mathbb{C}_+$, the modulus of the factor $\phi_m(0)/\phi_m(t)$ decreases as t moves from 0 toward $-\infty$. Moreover, $\hat{\mathbf{r}}_m(0)$ is precisely the residual of the sGMRES approximant $\hat{\mathbf{x}}_m(0)$. The following lemma provides a bound for $\|\hat{\mathbf{r}}_m(0)\|$.

LEMMA 5.3 (Bound on the residual). *Let $\mathbf{A} \in \mathbb{C}^{N \times N}$ be positive real, and let $\mathbf{b} \in \mathbb{C}^N$. Let $\hat{\mathbf{x}}_m(t)$ be the sHmArnoldi approximation to the solution of the shifted linear system (2.3), and let $\hat{\mathbf{r}}_m(t)$ be the corresponding residual defined in (5.2). Then*

$$\|\hat{\mathbf{r}}_m(0)\| \leq C_\varepsilon \alpha_m \|\mathbf{b}\|,$$

where $C_\varepsilon = \sqrt{(1+\varepsilon)/(1-\varepsilon)}$ and $\alpha_m = (1 - \delta\delta')^{m/2} \in [0, 1)$. The constants δ and δ' are defined by

$$\begin{aligned} \delta &= \min\{|\mathbf{v}^* \mathbf{A} \mathbf{v}| : \mathbf{v} \in \mathbb{C}^N, \|\mathbf{v}\| = 1\}, \\ \delta' &= \min\{|\mathbf{v}^* \mathbf{A}^{-1} \mathbf{v}| : \mathbf{v} \in \mathbb{C}^N, \|\mathbf{v}\| = 1\}. \end{aligned} \quad (5.3)$$

Proof. Let $\mathbf{x}_m(t)$ be the HmArnoldi approximation to the solution of the shifted linear system (2.3), and let $\mathbf{r}_m(t)$ be the corresponding residual. By [10, Lemma 6.4], $\|\mathbf{r}_m(0)\| \leq \alpha_m \|\mathbf{b}\|$. Since the subspace embedding property holds for the relevant residual vectors, by the sketched residual minimization property of sGMRES, we have

$$\|\hat{\mathbf{r}}_m(0)\| \leq \frac{1}{\sqrt{1-\varepsilon}} \|\mathbf{S}\hat{\mathbf{r}}_m(0)\| \leq \frac{1}{\sqrt{1-\varepsilon}} \|\mathbf{S}\mathbf{r}_m(0)\| \leq \sqrt{\frac{1+\varepsilon}{1-\varepsilon}} \|\mathbf{r}_m(0)\|.$$

Combining the above estimate with $\|\mathbf{r}_m(0)\| \leq \alpha_m \|\mathbf{b}\|$ completes the proof. $\square$

For a Stieltjes function f of the form (5.1), the sHmArnoldi approximation to $f(\mathbf{A})\mathbf{b}$ can be written as $\hat{\mathbf{f}}_m = \int_{-\infty}^0 g(t)\hat{\mathbf{x}}_m(t) dt$. Consequently,

$$\begin{aligned} f(\mathbf{A})\mathbf{b} - \hat{\mathbf{f}}_m &= \int_{-\infty}^0 g(t)(\mathbf{x}(t) - \hat{\mathbf{x}}_m(t)) dt \\ &= \int_{-\infty}^0 g(t)(t\mathbf{I} - \mathbf{A})^{-1}(\mathbf{b} - (t\mathbf{I} - \mathbf{A})\hat{\mathbf{x}}_m(t)) dt \\ &= \int_{-\infty}^0 g(t)(t\mathbf{I} - \mathbf{A})^{-1}\hat{\mathbf{r}}_m(t) dt \\ &= \int_{-\infty}^0 g(t)(t\mathbf{I} - \mathbf{A})^{-1}\eta_m(t)\hat{\mathbf{r}}_m(0) dt \\ &= \text{err}(\mathbf{A})\hat{\mathbf{r}}_m(0), \end{aligned}$$

where $\text{err}(z) = \int_{-\infty}^0 g(t)\eta_m(t)(t-z)^{-1} dt$. Combining the above discussion, we obtain the convergence of the restarted sHmArnoldi method for computing $f(\mathbf{A})\mathbf{b}$ when $\mathbf{A}$ is positive real and f is a Stieltjes function.

THEOREM 5.4 (Convergence of the restarted sHmArnoldi method). *Let $\mathbf{A} \in \mathbb{C}^{N \times N}$ be positive real, let $\mathbf{b} \in \mathbb{C}^N$, and let f be a Stieltjes function of the form (5.1). Assume that the sketched harmonic Ritz values satisfy $\theta_j^{[\ell]} \in \mathbb{C}_+$ for $0 \leq \ell \leq k$ and $1 \leq j \leq m$. Then the restarted sHmArnoldi approximation computed by Algorithm 5 satisfies*

$$\|f(\mathbf{A})\mathbf{b} - \hat{\mathbf{f}}^{[k]}\|_{\mathbf{A}^* \mathbf{A}} \leq C \|\mathbf{b}\| (C_\varepsilon \alpha_m)^{k+1}, \quad (5.4)$$

where $C = \sqrt{\nu_{\max}} f(\rho \nu_{\max})$, and α_m and C_ε are defined in Lemma 5.3. Here,

$$\begin{aligned} \nu_{\max} &= \max\{\mathbf{v}^* \mathbf{A}^* \mathbf{A} \mathbf{v} : \mathbf{v} \in \mathbb{C}^N, \|\mathbf{v}\| = 1\} = \|\mathbf{A}\|^2, \\ \rho &= \min\{\Re(\mathbf{v}^* \mathbf{A}^{-1} \mathbf{v}) : \mathbf{v} \in \mathbb{C}^N, \|\mathbf{v}\| = 1\}. \end{aligned} \quad (5.5)$$

Since $\mathbf{A}$ is positive real, we have $\rho > 0$. Thus, $\rho \nu_{\max} > 0$, and the representation (5.1) with g nonpositive implies that $f(\rho \nu_{\max}) > 0$.

Proof. By Lemma 5.2, the residual in the ℓ -th cycle satisfies $\hat{\mathbf{r}}_m^{[\ell]}(t) = \hat{\mathbf{r}}_m^{[\ell]}(0)\eta_m^{[\ell]}(t)$. Following the argument in Corollary 3.4, the remaining error after k restart cycles can be written as

$$f(\mathbf{A})\mathbf{b} - \hat{\mathbf{f}}^{[k]} = \left(\int_{-\infty}^0 g(t) \prod_{\ell=0}^k \eta_m^{[\ell]}(t) (t\mathbf{I} - \mathbf{A})^{-1} dt \right) \hat{\mathbf{r}}_m^{[k]}(0).$$

Since $\Re(\theta_j^{[\ell]}) > 0$, we have $\Re(-t/\theta_j^{[\ell]}) \geq 0$ for all $t \leq 0$. Hence, $|1 - t/\theta_j^{[\ell]}| \geq 1$, and

$$|\eta_m^{[\ell]}(t)| = \prod_{j=1}^m \frac{1}{|1 - t/\theta_j^{[\ell]}|} \leq 1, \quad 0 \leq \ell \leq k, \quad t \leq 0.$$

Following the proof of [10, Theorem 6.5] and using $|\eta_m^{[\ell]}(t)| \leq 1$, we obtain

$$\|f(\mathbf{A})\mathbf{b} - \hat{\mathbf{f}}^{[k]}\|_{\mathbf{A}^* \mathbf{A}} \leq \|\hat{\mathbf{r}}_m^{[k]}(0)\| \int_{-\infty}^0 g(t) \frac{\sqrt{\nu_{\max}}}{t - \rho\nu_{\max}} dt = \|\hat{\mathbf{r}}_m^{[k]}(0)\| \sqrt{\nu_{\max}} f(\rho\nu_{\max}).$$

Here, we used the fact that g is nonpositive in the Stieltjes representation (5.1). Finally, Lemma 5.3 gives $\|\hat{\mathbf{r}}_m^{[k]}(0)\| \leq C_\varepsilon \alpha_m \|\hat{\mathbf{r}}_m^{[k-1]}(0)\|$. Applying this inequality recursively and defining $C = \sqrt{\nu_{\max}} f(\rho\nu_{\max})$ completes the proof. $\square$

Theorem 5.4 states that, if the sketching matrix $\mathbf{S}$ provides a sufficiently accurate embedding for the subspace spanned by $\mathbf{B}_{m+1}$ in each restart cycle, then the restarted sHmArnoldi method converges geometrically. More precisely, this requires $C_\varepsilon \alpha_m < 1$ and the sketched harmonic Ritz values to satisfy $\theta_j^{[\ell]} \in \mathbb{C}_+$. Both conditions can be satisfied when the embedding distortion ε is sufficiently small. While C_ε depends explicitly on ε , the location of the sketched harmonic Ritz values is less transparent and also depends on the sketching matrix $\mathbf{S}$. The following subsection investigates this issue.

5.3. Sketched harmonic Ritz values. The convergence result in Theorem 5.4 assumes that all sketched harmonic Ritz values lie in the open right half-plane. This subsection provides a sufficient condition under which this assumption holds. The argument relies only on the orthogonality condition produced by the sHm-Arnoldi process, namely, $\mathbf{b}_{m+1} \perp_{\mathbf{S}} \mathbf{A}\mathbf{B}_m$.

Denote $\mathbf{W}_m = \mathbf{A}\mathbf{B}_m$. Since $\mathbf{b}_{m+1} \perp_{\mathbf{S}} \mathbf{A}\mathbf{B}_m$, multiplying the Arnoldi-like decomposition (2.4) from the left by $(\mathbf{S}\mathbf{W}_m)^*$ gives

$$(\mathbf{S}\mathbf{W}_m)^*(\mathbf{S}\mathbf{W}_m) = (\mathbf{S}\mathbf{W}_m)^*(\mathbf{S}\mathbf{B}_m)\mathbf{H}_m.$$

Therefore, $\mathbf{H}_m = \mathbf{G}_m^{-1}\mathbf{F}_m$, where $\mathbf{G}_m = (\mathbf{S}\mathbf{W}_m)^*(\mathbf{S}\mathbf{B}_m)$ and $\mathbf{F}_m = (\mathbf{S}\mathbf{W}_m)^*(\mathbf{S}\mathbf{W}_m)$, provided that $\mathbf{G}_m$ is nonsingular. Thus, the sketched harmonic Ritz values are the generalized eigenvalues of the matrix pencil $(\mathbf{F}_m, \mathbf{G}_m)$.

Let $\theta \in \Lambda(\mathbf{H}_m)$, and let $\mathbf{y} \in \mathbb{C}^m$ be a corresponding nonzero eigenvector. Set $\mathbf{x} = \mathbf{W}_m\mathbf{y}$. Then $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{F}_m\mathbf{y} = \mathbf{G}_m\mathbf{y}\theta$. This gives

$$\theta = \frac{\mathbf{y}^* \mathbf{F}_m \mathbf{y}}{\mathbf{y}^* \mathbf{G}_m \mathbf{y}} = \frac{\|\mathbf{S}\mathbf{x}\|^2}{(\mathbf{S}\mathbf{x})^*(\mathbf{S}\mathbf{A}^{-1}\mathbf{x})}.$$

Equivalently,

$$\frac{1}{\theta} = \frac{(\mathbf{S}\mathbf{x})^*(\mathbf{S}\mathbf{A}^{-1}\mathbf{x})}{\|\mathbf{S}\mathbf{x}\|^2}. \quad (5.6)$$

Taking $\mathbf{u} = \mathbf{A}^{-1}\mathbf{x} = \mathbf{B}_m\mathbf{y}$ and $\mathbf{v} = \mathbf{x} = \mathbf{A}\mathbf{B}_m\mathbf{y}$ in the inner-product form of the subspace embedding property (2.10), we obtain

$$|(\mathbf{S}\mathbf{x})^*(\mathbf{S}\mathbf{A}^{-1}\mathbf{x}) - \mathbf{x}^* \mathbf{A}^{-1} \mathbf{x}| \leq \varepsilon \|\mathbf{A}^{-1} \mathbf{x}\| \|\mathbf{x}\| \leq \varepsilon \|\mathbf{A}^{-1}\| \|\mathbf{x}\|^2.$$

Combining this estimate with the subspace embedding property (2.9) and the identity (5.6), we conclude that the reciprocal of every sketched harmonic Ritz value satisfies

$$\frac{1}{\theta} \in \left\{ \frac{z_1 + z_2}{\tau} : z_1 \in \mathcal{W}(\mathbf{A}^{-1}), |z_2| \leq \varepsilon \|\mathbf{A}^{-1}\|, 1 - \varepsilon \leq \tau \leq 1 + \varepsilon \right\}.$$

Since $\mathbf{A}$ is positive real, $\mathbf{A}^{-1}$ is also positive real. Let ρ be defined as in (5.5). If $\varepsilon \|\mathbf{A}^{-1}\| < \rho$, then, for any sketched harmonic Ritz value θ , the preceding inclusion gives

$$\Re\left(\frac{1}{\theta}\right) \geq \frac{\rho - \varepsilon \|\mathbf{A}^{-1}\|}{1 + \varepsilon} > 0.$$

Since $\Re(1/\theta) > 0$ is equivalent to $\Re(\theta) > 0$, all sketched harmonic Ritz values lie in the open right half-plane. Moreover, under the condition $\varepsilon \|\mathbf{A}^{-1}\| < \rho$, for every nonzero $\mathbf{y} \in \mathbb{C}^m$, with $\mathbf{x} = \mathbf{A}\mathbf{B}_m\mathbf{y}$, we have

$$\Re(\mathbf{y}^* \mathbf{G}_m \mathbf{y}) = \Re((\mathbf{S}\mathbf{x})^* (\mathbf{S}\mathbf{A}^{-1}\mathbf{x})) \geq (\rho - \varepsilon \|\mathbf{A}^{-1}\|) \|\mathbf{x}\|^2 > 0.$$

Hence, $\mathbf{G}_m$ is nonsingular, and the condition $\varepsilon \|\mathbf{A}^{-1}\| < \rho$ is sufficient for the sketched harmonic Ritz values generated by the sHm-Arnoldi process to lie in $\mathbb{C}_+$, as required in Theorem 5.4.

6. Numerical experiments. In this section, we present numerical experiments to demonstrate the stability and efficiency of the proposed sketch-and-restart algorithms. The experiments illustrate the computational savings achieved through sketching, the storage reduction obtained by the adaptive t -Arnoldi process, and the effectiveness of thick restarting when aggressive truncation slows convergence. All algorithms are implemented in MATLAB R2023b, and all experiments are performed on a server equipped with two Intel Xeon Gold 6226R CPUs running at 2.90 GHz and 1000.6 GB of RAM. The code for reproducing the numerical experiments is available at https://github.com/JingyuLiuMath/funm_quad.

We use a sparse sign matrix [15] as the sketching matrix in all examples. Specifically, the matrix is defined as

$$\mathbf{S} = \sqrt{\frac{1}{\zeta}} [\mathbf{s}_1 \quad \cdots \quad \mathbf{s}_N] \in \mathbb{C}^{s \times N},$$

where ζ is the sparsity parameter. Each vector $\mathbf{s}_i$ is constructed by selecting ζ entries uniformly at random and assigning them the values ± 1 with equal probability. Larger values of ζ generally improve the embedding quality but increase the computational cost, whereas smaller values may lead to instability. Following the recommendation in [15], we set $\zeta = \min\{s, 8\}$. When the Krylov subspace dimension m is fixed, we set the sketching dimension to $s = 2m$. Applying this sketching matrix to a vector therefore requires only $\mathcal{O}(N)$ operations.

The quadrature tolerance and convergence tolerance are set to 10^{-9} and 10^{-7} , respectively, and the maximum number of quadrature points is set to 512 in all examples. The relative error is computed using the exact solution. For Algorithm 5, we denote the variants based on the s-Arnoldi and sHm-Arnoldi processes by sArn and sHmArn, respectively. For comparison, we also use the quadrature-based restarted Arnoldi method of [11], which generates an orthonormal basis using the standard

Arnoldi process and computes the Arnoldi approximation in each restart cycle. We denote this method by Arn.

For Algorithm 6, the initial maximum Krylov subspace dimension is set to $m_{\max} = m$. In the t -Arnoldi process, we use $t \in \{2, 1, 0\}$, the condition number threshold $\tau = 10^6$, the sketching dimension increment $s_0 = 30$, and the sketching dimension control parameter $\eta = 2$. At the end of each t -Arnoldi process, a rank-1 update is applied to enforce either $\mathbf{b}_{m+1} \perp \mathbf{B}_m$ or $\mathbf{b}_{m+1} \perp \mathbf{A}\mathbf{B}_m$. We denote the corresponding methods by adaArn- t and adaHmArn- t , respectively.

Finally, we use n_{iter} , n_{matvec} , and n_{vec} to denote the number of restart cycles, the total number of matrix-vector products, and the maximum number of vectors stored simultaneously, respectively.

6.1. Convection-diffusion example. In this example, $\mathbf{A}$ is obtained by discretizing a two-dimensional convection-diffusion operator on the unit square with a constant convection field in the direction $[1, -1]$ and diffusion coefficient $D = 10^{-3}$. More precisely,

$$\mathbf{A} = \frac{D}{h^2}(\mathbf{L} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{L}) + \frac{1}{h}(\mathbf{C} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{C}^\top) \in \mathbb{R}^{n^2 \times n^2},$$

where $h = 1/(n+1)$, $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the tridiagonal matrix with diagonal entries 2 and sub- and superdiagonal entries -1 , and $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the matrix with diagonal entries 1 and subdiagonal entries -1 . We choose $n = 100$, so that $N = 10,000$, and compute $f(\mathbf{A})\mathbf{b}$ for $f(z) = z^{-1/2}$, where $\mathbf{b}$ is the vector of all ones normalized to unit norm. The Krylov subspace dimension is set to $m = 150$.

method	error	time (s)	n_{iter}	n_{matvec}	n_{vec}
Arn	1.5e-08	1.7e+00	3	450	150
sArn	3.1e-09	1.2e+00	4	600	150
sHmArn	3.9e-09	1.2e+00	4	600	150
adaArn-2	1.5e-08	1.3e+00	3	450	150
adaArn-1	4.4e-08	8.3e-01	5	420	90
adaArn-0	2.8e-08	4.1e-01	19	263	15
adaHmArn-2	4.5e-08	1.2e+00	3	450	150
adaHmArn-1	1.4e-08	7.3e-01	4	355	90
adaHmArn-0	3.3e-08	4.1e-01	18	251	15

Table 6.1: Relative error, runtime, number of restart cycles, number of matrix-vector products, and maximum number of stored vectors for the inverse square root function in the convection-diffusion example.

Table 6.1 and Figure 6.1 show that all methods attain the prescribed accuracy, with final relative errors of the same order of magnitude. Among the fixed-dimension methods, sArn and sHmArn require four restart cycles and 600 matrix-vector products, compared with three cycles and 450 matrix-vector products for Arn. Nevertheless, both sketched methods reduce the runtime from 1.7 seconds to 1.2 seconds. Thus,

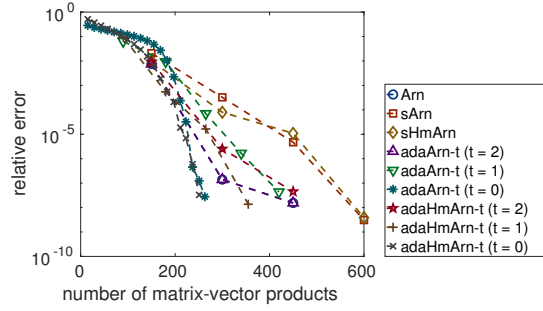


Fig. 6.1: Relative error of the approximation versus the number of matrix-vector products in the convection-diffusion example.

the lower cost of orthogonalization in the sketched space more than compensates for the additional restart cycle and matrix-vector products.

For both adaptive variants, decreasing the truncation parameter t substantially reduces the storage requirement, from 150 vectors for $t = 2$ to 90 for $t = 1$ and 15 for $t = 0$. In this example, more aggressive truncation also reduces both the runtime and the total number of matrix-vector products. In particular, `adaArn-0` and `adaHmArn-0` require only 263 and 251 matrix-vector products, respectively, and both complete in 0.41 seconds while maintaining relative errors below the convergence tolerance. The results therefore indicate that, for this problem, aggressive truncation provides substantial savings in both storage and computational cost, despite increasing the number of restart cycles.

6.2. Lattice quantum chromodynamics example. This example arises from lattice quantum chromodynamics (QCD), where one needs to solve linear systems involving the overlap Dirac operator [18],

$$N_{\text{ovl}} = \rho \mathbf{I} + \gamma_5 \text{sign}(\mathbf{Q}).$$

Here, $\rho > 1$, γ_5 is a permutation matrix, and $\mathbf{Q}$ is a large sparse matrix that becomes non-Hermitian in the presence of a nonzero chemical potential. Since forming $\text{sign}(\mathbf{Q})$ explicitly is infeasible, its action on a vector is computed as

$$\text{sign}(\mathbf{Q})\mathbf{b} = (\mathbf{Q}^2)^{-1/2}\mathbf{Q}\mathbf{b}.$$

In our experiment, $N = 49,152$, $\mathbf{b} = \mathbf{e}_1$, and $m = 150$.

Table 6.2 and Figure 6.2 show that all methods attain the prescribed accuracy. The behavior in this example differs from the convection-diffusion problem. Among the fixed-dimension methods, `sArn` and `sHmArn` require the same number of restart cycles and matrix-vector products as `Arn`, but reduce the runtime from 14 seconds to 13 seconds. This again illustrates the lower per-cycle cost of the sketched orthogonalization process.

For the adaptive methods, $t = 2$ provides the best overall performance in this example. The methods `adaArn-2` and `adaHmArn-2` store at most 105 vectors, compared with 150 for the fixed-dimension methods, while requiring only 410 and 415 matrix-vector products, respectively. Their runtimes are both 12 seconds, which are the smallest among all methods considered. Further decreasing t substantially reduces

method	error	time (s)	n_{iter}	n_{matvec}	n_{vec}
Arn	4.7e-10	1.4e+01	3	450	150
sArn	1.3e-09	1.3e+01	3	450	150
sHmArn	1.6e-09	1.3e+01	3	450	150
adaArn-2	7.8e-09	1.2e+01	4	410	105
adaArn-1	9.9e-08	1.5e+01	40	722	20
adaArn-0	9.1e-08	1.9e+01	106	964	10
adaHmArn-2	1.2e-08	1.2e+01	4	415	105
adaHmArn-1	9.6e-08	1.6e+01	41	740	20
adaHmArn-0	9.8e-08	2.8e+01	142	1279	10

Table 6.2: Relative error, runtime, number of restart cycles, number of matrix-vector products, and maximum number of stored vectors for the inverse square root function in the QCD example.

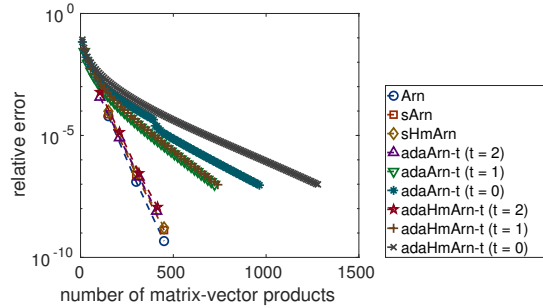


Fig. 6.2: Relative error of the approximation versus the number of matrix-vector products in the QCD example.

the storage requirement, to 20 vectors for $t = 1$ and 10 vectors for $t = 0$, but leads to a pronounced deterioration in convergence. In particular, the number of restart cycles increases to 40–41 for $t = 1$ and to 106–142 for $t = 0$, accompanied by a corresponding increase in the number of matrix-vector products and runtime. Thus, for this QCD example, $t = 2$ offers the best balance among storage, matrix-vector products, and runtime.

We also apply thick restarting [9] to sArn and the adaptive Arnoldi-type methods, retaining the Ritz vectors associated with the 5 Ritz values closest to the origin. As shown in Figure 6.3, thick restarting has little effect on sArn, whose convergence is already fast; it slightly increases the runtime from 12.7 to 13.5 seconds. In contrast, thick restarting accelerates all three adaptive methods, with increasingly pronounced benefits as t decreases. For adaArn-2, the runtime decreases from 12.2 to 10.3 seconds. The improvement is more substantial for adaArn-1 and adaArn-0, whose runtimes decrease from 15.4 to 8.8 seconds and from 19.1 to 10.5 seconds, respectively. These results show that thick restarting effectively mitigates the slower convergence caused

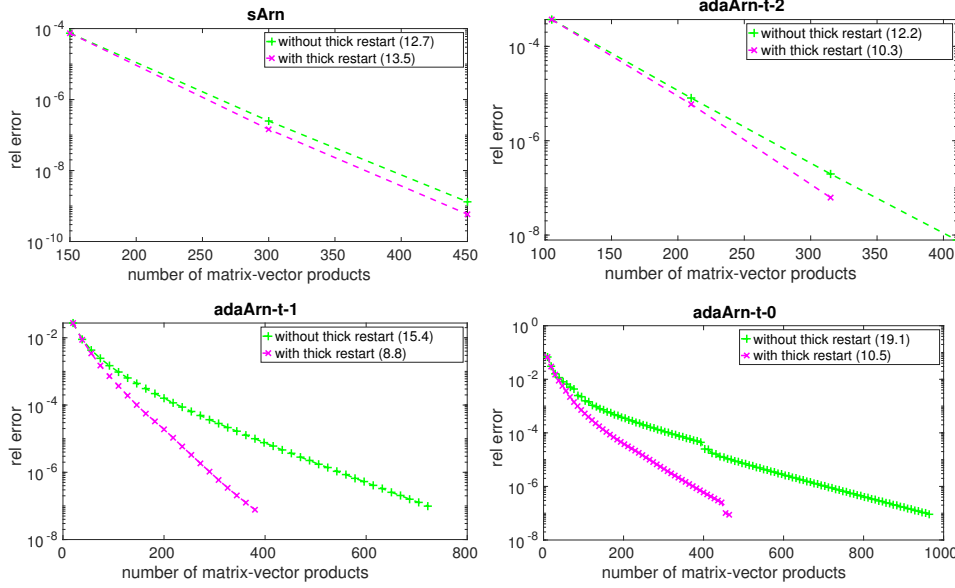


Fig. 6.3: Thick restarting on the QCD example. The runtime of each method is also shown in the legend.

by aggressive truncation while retaining the storage savings of the adaptive methods.

6.3. Fractional graph Laplacian example. In this example, $\mathbf{A} \in \mathbb{R}^{6301 \times 6301}$ is the nonsymmetric adjacency matrix of the `p2p_Gnutella08` network from the SuiteSparse Matrix Collection [6]. We define the in-degree graph Laplacian by $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The matrix $\mathbf{L}$ is singular, and its spectrum lies in the closed right half-plane. We interpret the inverse square root as acting on the invariant subspace associated with the nonzero spectrum and compute

$$\mathbf{L}^{1/2} \mathbf{b} = \mathbf{L}^{-1/2} \mathbf{L} \mathbf{b}.$$

Since $\mathbf{L} \mathbf{b} \in \mathcal{R}(\mathbf{L})$, the Krylov subspace methods considered here can be applied to this inverse-square-root action. We set $m = 100$.

Table 6.3 and Figure 6.4 show that all methods attain the prescribed accuracy. Among the fixed-dimension methods, sArn and sHmArn require the same number of restart cycles and matrix-vector products as Arn, but reduce the runtime from 0.84 seconds to 0.43 seconds. Thus, in this example, sketching nearly halves the computational time without affecting the convergence behavior.

Among the adaptive methods, $t = 2$ provides the best overall performance. The methods adaArn-2 and adaHmArn-2 store at most 45 vectors, compared with 100 for the fixed-dimension methods, while requiring 355 and 395 matrix-vector products, respectively. Their runtimes, 0.44 and 0.41 seconds, are comparable to those of sArn and sHmArn. Further decreasing t reduces the storage requirement to 30 vectors for $t = 1$ and 15 vectors for $t = 0$, but substantially slows convergence. In particular, both the number of restart cycles and the number of matrix-vector products increase markedly as t decreases. Hence, for this example, $t = 2$ offers the best compromise among storage reduction, convergence speed, and runtime.

method	error	time (s)	n_{iter}	n_{matvec}	n_{vec}
Arn	4.3e-08	8.4e-01	3	300	100
sArn	4.3e-08	4.3e-01	3	300	100
sHmArn	4.3e-08	4.3e-01	3	300	100
adaArn-2	8.0e-08	4.4e-01	8	355	45
adaArn-1	5.5e-08	6.2e-01	33	606	30
adaArn-0	9.9e-08	8.3e-01	78	785	15
adaHmArn-2	6.6e-08	4.1e-01	9	395	45
adaHmArn-1	9.2e-08	4.7e-01	25	495	30
adaHmArn-0	8.7e-08	8.1e-01	74	745	15

Table 6.3: Relative error, runtime, number of restart cycles, number of matrix-vector products, and maximum number of stored vectors for the inverse square root function in the fractional graph Laplacian example.

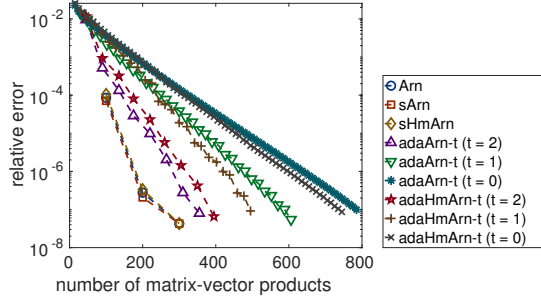


Fig. 6.4: Relative error of the approximation versus the number of matrix-vector products in the fractional graph Laplacian example.

We also test thick restarting with 5 retained Ritz vectors. As shown in Figure 6.5, thick restarting improves both convergence and runtime for all displayed methods. For sArn, the number of matrix-vector products is reduced from 300 to 200, and the runtime decreases from 0.4 to 0.3 seconds. For the adaptive methods, the benefits become more pronounced as t decreases. For adaArn-2, thick restarting reduces the number of matrix-vector products from approximately 360 to 210 and the runtime from 0.4 to 0.3 seconds. For adaArn-1, the corresponding reductions are from approximately 600 to 210 matrix-vector products and from 0.6 to 0.3 seconds, while for adaArn-0 they are from approximately 800 to 260 matrix-vector products and from 0.8 to 0.6 seconds. These results show that thick restarting effectively compensates for the slower convergence caused by aggressive truncation while preserving the storage advantage of the adaptive methods.

7. Conclusions. We have developed a sketch-and-restart framework for computing $f(\mathbf{A})\mathbf{b}$ with large sparse non-Hermitian matrices. The framework combines

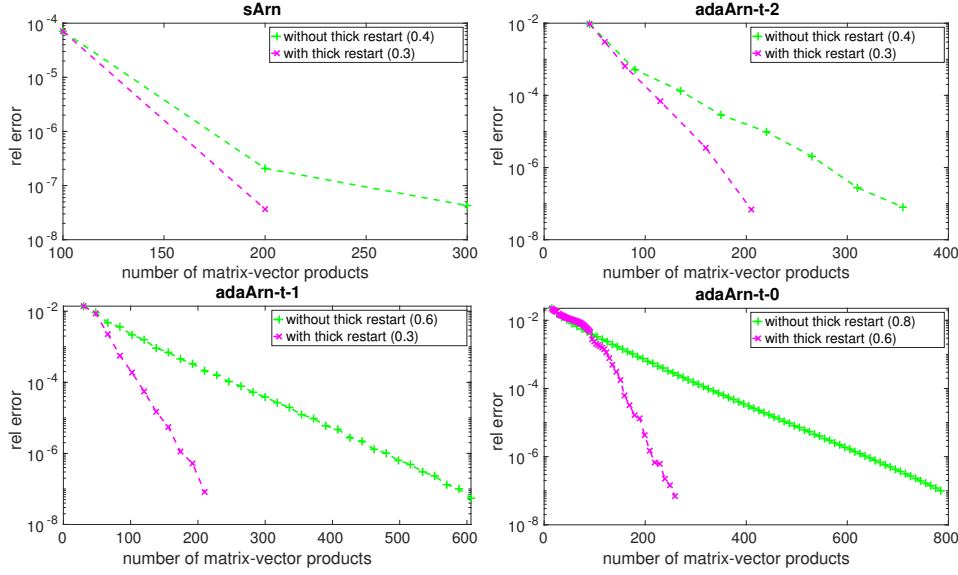


Fig. 6.5: Thick restarting on the fractional graph Laplacian example. The runtime of each method is shown in the legend.

quadrature-based restarting with Arnoldi-like decompositions generated by sketched or truncated Krylov processes. The key observation is that several Krylov and sketched Krylov approximations share the common form $\mathbf{B}_m f(\mathbf{H}_m) \mathbf{e}_1 \beta$, which leads to a unified error representation and restarting mechanism. Within this framework, we combined the existing s-Arnoldi process with quadrature-based restarting and introduced a new sHm-Arnoldi process together with its restarted variant. We further developed adaptive restarted Arnoldi and HmArnoldi methods based on the t -Arnoldi process and a rank-1 update. For Stieltjes functions, we established the convergence of the restarted sHmArnoldi method when $\mathbf{A}$ is positive real, together with a sufficient condition for the required location of the sketched harmonic Ritz values. More broadly, the proposed framework separates the restarting mechanism from the basis generation procedure, making it possible to incorporate other randomized Krylov processes within the same framework.

Numerical experiments demonstrate the stability and efficiency of the proposed methods. The sketched methods consistently reduce the runtime by lowering the cost of orthogonalization, while the adaptive methods avoid prescribing the restart length m a priori and can substantially reduce storage. The experiments also demonstrate that thick restarting is an effective complement to aggressive truncation, recovering much of the resulting loss in convergence while preserving the associated storage savings. In particular, the case $t = 0$ is promising because the inner products required for the rank-1 update can be computed simultaneously at the BLAS-2 level, suggesting that the proposed methods are well suited for high-performance implementations.

Acknowledgments. JL thanks the China Scholarship Council (CSC) for supporting his visit to The University of Manchester. SG and LN acknowledge funding from the UK's Engineering and Physical Sciences Research Council (EPSRC grant EP/Z533786/1). SG is supported by Royal Society Industry Fellowship IF/R1/231032.

REFERENCES

- [1] M. AFANASJEW, M. EIERMANN, O. G. ERNST, AND S. GÜTTEL, *Implementation of a restarted Krylov subspace method for the evaluation of matrix functions*, Linear Algebra Appl., 429 (2008), pp. 2293–2314.
- [2] O. BALABANOV AND L. GRIGORI, *Randomized Gram–Schmidt process with application to GMRES*, SIAM J. Sci. Comput., 44 (2022), pp. A1450–A1474.
- [3] B. BECKERMANN AND A. TOWNSEND, *On the singular values of matrices with displacement structure*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1227–1248.
- [4] L. BURKE AND S. GÜTTEL, *Krylov subspace recycling with randomized sketching for matrix functions*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 2243–2262.
- [5] A. CORTINOVIS, D. KRESSNER, AND Y. NAKATSUKASA, *Speeding up Krylov subspace methods for computing $f(\mathbf{A})\mathbf{b}$ via randomization*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 619–633.
- [6] T. A. DAVIS AND Y. HU, *The University of Florida sparse matrix collection*, ACM Trans. Math. Softw., 38 (2011), pp. 1:1–1:25.
- [7] J.-G. DE DAMAS, L. GRIGORI, I. SIMUNEC, AND E. TIMSIT, *Randomized orthogonalization and Krylov subspace methods: Principles and algorithms*, arXiv:2512.15455, (2025).
- [8] M. EIERMANN AND O. G. ERNST, *A restarted Krylov subspace method for the evaluation of matrix functions*, SIAM J. Numer. Anal., 44 (2006), pp. 2481–2504.
- [9] M. EIERMANN, O. G. ERNST, AND S. GÜTTEL, *Deflated restarting for matrix functions*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 621–641.
- [10] A. FROMMER, S. GÜTTEL, AND M. SCHWEITZER, *Convergence of restarted Krylov subspace methods for Stieltjes functions of matrices*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1602–1624.
- [11] 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.
- [12] L. GRIGORI, D. KRESSNER, N. SHAO, AND I. SIMUNEC, *Restoring similarity in randomized Krylov methods with applications to eigenvalue problems and matrix functions*, arXiv:2601.10248, (2026).
- [13] 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.
- [14] N. J. HIGHAM, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, PA, 2008.
- [15] P.-G. MARTINSSON AND J. A. TROPP, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numer., 29 (2020), pp. 403–572.
- [16] R. B. MORGAN, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.
- [17] 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.
- [18] H. NEUBERGER, *Exactly massless quarks on the lattice*, Phys. Lett., B, 417 (1998), pp. 141–144.
- [19] C. C. PAIGE, B. N. PARLETT, AND H. A. VAN DER VORST, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–133.
- [20] Y. SAAD, *Iterative Methods for Sparse Linear Systems, 2nd ed.*, SIAM, Philadelphia, PA, 2003.
- [21] D. P. WOODRUFF, *Sketching as a tool for numerical linear algebra*, Found. Trends Theor. Comput. Sci., 10 (2014), pp. 1–157.