On an Economic Arnoldi Method for BML Matrices

Bernhard Beckermann, Clara Mertens, Raf Vandebril

Bernhard Beckermann, Clara Mertens, Raf Vandebril. On an Economic Arnoldi Method for BML Matrices. SIAM Journal on Matrix Analysis and Applications, 2018, 39 (2), pp.737-768. 10.1137/17M1114363 . hal-01451722

On a fast Arnoldi method for BML-matrices

Bernhard Beckermann, Clara Mertens, and Raf Vandebril

Abstract

Matrices whose adjoint is a low rank perturbation of a rational function of the matrix naturally arise when trying to extend the well known Faber-Manteuffel theorem [7, 8], which provides necessary and sufficient conditions for the existence of a short Arnoldi recurrence. We show that an orthonormal Krylov basis for this class of matrices can be generated by a short recurrence relation based on GMRES residual vectors. These residual vectors are computed by means of an updating formula. Furthermore, the underlying Hessenberg matrix has an accompanying low rank structure, which we will investigate closely.

1 Introduction

In this article we will discuss a new variant of the Arnoldi method applied to a class of sparse matrices $A \in \mathbb{C}^{n\times n}$ which allows to compute the first k Arnoldi vectors in complexity $\mathcal{O}(kn)$. We will refer to this class of matrices as *BML-matrices*, following the fundamental work of Barth & Manteuffel [2] and Liesen [13] on matrices A whose adjoint is a low rank perturbation of a rational function of A. More specifically, we assume that

$$A^* = p(A)q(A)^{-1} + FG^*, \tag{1.1}$$

with p, q polynomials of degree m_1 and m_2, respectively, and

$$F = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{m_3}], \quad G = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{m_3}] \in \mathbb{C}^{n\times m_3}$$

matrices of full column rank. Moreover, it is assumed that the roots of q are simple. By taking $p(z)/q(z) \in \{z, 1/z\}$, we see that Hermitian matrices and unitary matrices are BML-matrices, and the same is true for low rank perturbations of such matrices. Furthermore, if $FG^* = 0$ in (1.1), the matrix A is normal [13]. In what follows we suppose that $m_j \ll n$, since these quantities (as well as the sparsity pattern of A) are hidden in the constant of the above-claimed complexity result.

After k steps of the Arnoldi process with initial vector $\mathbf{b}$ one obtains the expression

$$AV_k = V_k H_k + h_{k+1,k}\mathbf{v}_{k+1}\mathbf{e}_k^*, \tag{1.2}$$

with $V_k = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$, $\mathbf{v}_1 = \mathbf{b}/\|\mathbf{b}\|$, $\mathbf{v}_{k+1} \in \mathbb{C}^n$ and $h_{k+1,k} \geq 0$ satisfying $V_k^* V_k = I_k$ and $V_k^* \mathbf{v}_{k+1} = 0$. The matrix H_k is upper Hessenberg.

A fast variant of the Arnoldi process will exploit additional structure of the upper Hessenberg matrix H_k. For example, to compute the successive vectors $\mathbf{v}_k$ for Hermitian A we get a tridiagonal H_k and the Arnoldi process reduces to the Lanczos method. For matrices A satisfying (1.1), H_k turns out to be a rank structured matrix. In order to specify this statement, the following definition is introduced.

Definition 1.1. *We say that a matrix $B \in \mathbb{C}^{n\times n}$ is $(r, s)-$upper-separable with $s \geq 0$, if for all $j = 1, 2, \dots, n - |r|$ it holds that (in Matlab notation)*

$$\textit{rank } B(1 : j - \min(r, 0), j + \max(r, 0) : n) \leq s.$$

In other words, any submatrix of B including elements on and above the rth diagonal of B is of rank at most s. The 0th diagonal corresponds to the main diagonal, while the rth diagonal refers to the rth superdiagonal if $r > 0$ and to the $-r$th subdiagonal if $r < 0$.

Before proceeding, we give some comments on Definition 1.1 whose formulation is inspired by some related well-established definitions. For example, matrices with both B and B^* being $(0,1)$–upper-separable (and $(1,1)$–upper-separable, respectively) are referred to as semiseparable matrices (and quasi separable, respectively) [18, §1.1, §9.3.1]. As a simple example, a tridiagonal matrix is quasi separable, and its inverse is known to be semiseparable. Matrices being $(1-s,s)$–upper-separable with their adjoint being $(1-r,r)$–upper-separable are usually called $(r,s)-$semiseparable, while matrices being $(1,s)$–upper-separable with their adjoint being $(1,r)$–upper-separable are called $(r,s)-$quasi separable [18, §8.2.2 and §8.2.3].

The above statement on the rank structure of H_k can now be made exact. Assume the Arnoldi process breaks down after $N \leq n$ iterations, i.e., $\mathbf{v}_{N+1} = 0$ in (1.2). We will refer to N as the 'Arnoldi termination index' in the rest of the article. In Corollary 3.2 it is shown that for a BML-matrix A, the underlying Hessenberg matrix H_N is $(r,s)-$upper-separable, where r and s are functions of m_j. However, for a general matrix A, there is no reason for the underlying Hessenberg matrix to be upper-separable. The upper-separable structure of the underlying Hessenberg matrix gives rise to the design of a *multiple recurrence relation* [2,14], signifying that each new Arnoldi vector can be written as

$$\mathbf{v}_{k+1} = \sum_{j=k-m_3-m_2}^{k} \alpha_{k,j} A\mathbf{v}_j + \sum_{j=k-m_3-m_1}^{k} \beta_{k,j}\mathbf{v}_j. \tag{1.3}$$

The smaller the quantities m_j in (1.1), the shorter the recurrence relation becomes. In [5] the same recurrence relations are derived for the class of so called (H,m)-well-free matrices. We refer to Remark 3.7 for some more details.

In this article we investigate a different version of the recurrence relation (1.3) by rewriting it in terms of GMRES residual vectors, aiming to overcome some of the numerical problems which relation (1.3) entails, such as the possibility of a breakdown [14]. It will be shown how these GMRES residual vectors can be computed progressively by means of an updating formula. This partially extends the discussion on a progressive GMRES method for nearly Hermitian matrices as presented by Beckermann & Reichel [4].

The article is organized as follows. Section 2 describes the structure of the unitary factor Q in the QR-decomposition of a Hessenberg matrix in terms of orthogonal polynomials (the results in this section are valid for a general matrix $A \in \mathbb{C}^{n\times n}$). It is well known that this unitary factor can be represented as a product of Givens rotations, see e.g., the isometric Arnoldi process introduced by Gragg [10] and its extension to the class of shifted unitary matrices by Jagels & Reichel [12]. We will describe these Givens rotations by means of orthogonal polynomials. Furthermore, links between orthogonal polynomials and the GMRES algorithm are discussed, leading to an updating formula to compute the GMRES residuals progressively. In section 3 the upper-separable structure of the Hessenberg matrix related to the Arnoldi process applied to a BML-matrix is investigated. It is shown how this upper-separable structure can be generated by the GMRES residual vectors, allowing to construct a short recurrence relation and an accompanying algorithm. In section 4 we compare our findings with those presented in [2]. Section 5 discusses some computational reductions that can be made in case the matrix A is nearly unitary or nearly shifted unitary. Finally, section 6 discusses the numerical performance and stability of the algorithm.

Throughout this article we will make use of the following notation. Vectors are written in bold face lower case letters, e.g., $\mathbf{x}, \mathbf{y}$, and $\mathbf{z}$. The vector $\mathbf{e}_k$ denotes the kth column of an identity matrix of applicable order. The standard inner product is denoted as $\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{y}^*\mathbf{x}$, with $\cdot^*$ the Hermitian conjugate. We write $\|\cdot\|$ for the induced Euclidean norm as well as the subordinate spectral matrix norm. Matrices are denoted by upper case letters $A = (a_{ij})$, and I_k denotes the identity matrix of order k. We will frequently express formula (1.2) in the form

$$AV_k = V_{k+1}\underline{H}_k, \quad \text{with} \quad V_{k+1} := [V_k, \mathbf{v}_{k+1}],$$

where

$$\underline{I}_k := \begin{bmatrix} I_k \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}, \quad \underline{H}_k := \begin{bmatrix} H_k \\ h_{k+1,k}\mathbf{e}_k^* \end{bmatrix} = H_{k+1}\underline{I}_k \in \mathbb{C}^{(k+1)\times k},$$

and $V_k = V_{k+1}\underline{I}_k$, revealing the nested structure of the Arnoldi matrices V_k and H_k.

2 Towards a progressive GMRES method

Many Krylov space methods and in particular the Arnoldi method can be described in polynomial language which reveals some particular properties, and makes the link to the rich theory of orthogonal polynomials. For example, from (1.2) one deduces by recurrence on the degree that, for any polynomial p

$$p(A)\mathbf{v}_1 = V_k p(H_k)\mathbf{e}_1, \quad \text{provided that } \deg p < k, \tag{2.1}$$

illustrating that Krylov spaces are intimately related to polynomials. See e.g., [16, §6.6.2] for the case of Hermitian A. However, polynomial language can be used also for general matrices [3, §1.3]. In §2.1 we will see that the Arnoldi vectors correspond to a (finite) family of polynomials which are orthogonal with respect to

$$\langle p, q\rangle_{A,\mathbf{v}_1} = \langle p(A)\mathbf{v}_1, q(A)\mathbf{v}_1\rangle = (q(A)\mathbf{v}_1)^*\, p(A)\mathbf{v}_1, \tag{2.2}$$

a scalar product on the set of polynomials of degree $< N$, with N the termination index of the Arnoldi method. The scalar product (2.2) induces a norm $\|\cdot\|_{A,\mathbf{v}_1}$. This implies in particular the well known fact [16, Proposition 6.7] that Arnoldi vectors are normalized FOM residuals. In §2.2 we will use polynomial language to give an explicit expression for the Q-factor in a QR-decomposition of an upper Hessenberg matrix. Such a formula for a unitary upper Hessenberg matrix is not known in literature, though of course there is a close link with Gragg's explicit formula in terms of Givens rotations [10,16, §6.5.3]. This formula will enable us to deduce in Corollary 2.4 a decay property of the entries of Q far from the main diagonal, and reveals immediately a rank structure for Q. In §2.3 we recall that GMRES residuals can be expressed in terms of orthogonal polynomials: we are faced with a well-studied extremal problem for general orthogonal polynomials. This allows us in §2.4 to establish a well known link between (normalized) FOM and GMRES residuals [16, §6.5.5], allowing for a recursive computation of (normalized) GMRES residuals. Such a progressive GMRES implementation has been discussed before [16, §6.5.3] and [4]. The implementation presented in this article is inspired by the work of Beckermann & Reichel [4]. Both implementations make use of the decomposition of Q into a product of Givens rotations, but differ in how to find the angles of these rotations. We will consider in §2.4 not only FOM and GMRES for systems of linear equations $A\mathbf{x} = \mathbf{b}$ but more generally for shifted systems $(A - \delta I)\mathbf{x} = \mathbf{b}$ for some parameters $\delta \in \mathbb{C}$. All findings of this section hold for general matrices A.

2.1 Orthogonal polynomials linked to the Arnoldi process

Given an $N \times N$ upper Hessenberg matrix H_N with positive real entries $h_{k+1,k}$ on the subdiagonal. We define polynomials $q_0, q_1, ..., q_{N-1}$ recursively through the formula

$$\begin{aligned} q_0(z) &= 1, \\ q_k(z)h_{k+1,k} &= zq_{k-1}(z) - \sum_{j=1}^{k} q_{j-1}(z)h_{j,k}, \text{ if } N-1 \geq k \geq 1. \end{aligned} \tag{2.3}$$

Direct computation yields the following well known link with the characteristic polynomials of the principal submatrices:

$$q_k(z) = \frac{1}{\prod_{i=1}^{k} h_{i+1,i}} \det(zI_k - H_k), \tag{2.4}$$

showing that q_k is of degree k, with positive leading coefficient. In what follows we will write (2.3) in the form

$$(q_0(z), \ldots, q_k(z))\underline{H}_k = z(q_0(z), \ldots, q_{k-1}(z)), \tag{2.5}$$

where H_k is the $k \times k$ principal minor of H_N and $N-1 \geq k \geq 1$. One deduces from (2.3) by recurrence on k that $q_k(H_N)\mathbf{e}_1 = \mathbf{e}_{k+1}$. Assume we apply the Arnoldi process to a matrix $A \in \mathbb{C}^{n\times n}$, that $N \leq n$ is the Arnoldi termination index and H_N the underlying Hessenberg matrix. Using (2.1), this implies

$$\mathbf{v}_k = q_{k-1}(A)\mathbf{v}_1, \tag{2.6}$$

and thus indeed the q_j is the jth orthonormal polynomial with respect to the scalar product (2.2). Also, from (2.4) we see that the kth FOM iterate for the shifted system $(A - \delta I)\mathbf{x} = \mathbf{b}$ exists if and only if $q_k(\delta) \neq 0$, and that in this case the kth FOM residual is given by $q_k(A)\mathbf{b}/q_k(\delta)$, compare with [16, Proposition 6.7].

2.2 The QR-factorization of a Hessenberg matrix

We will derive an explicit formula for the unitary factor in the QR-decomposition of the upper Hessenberg matrix $\underline{H}_k$ in terms of the orthonormal polynomials $q_0, ..., q_k$. To our knowledge, such a result is new. It could also be potentially useful for studying the convergence of the QR-method with shifts.

Let $Q_{k+1}(\delta)$ be the unitary factor in the QR-decomposition of $\underline{H}_k - \delta \underline{I}_k$, i.e.,

$$Q_{k+1}(\delta)^*(\underline{H}_k - \delta \underline{I}_k) = \underline{R}_k(\delta) := \begin{bmatrix} R_k(\delta) \\ 0 \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}, \tag{2.7}$$

$R_k(\delta)$ an upper triangular matrix with positive real entries on its main diagonal. It is well known (see, e.g., [16, Subsection 6.5.3]) that $Q_{k+1}(\delta)$ can be obtained as a product of Givens rotations, which are applied to $\underline{H}_k - \delta \underline{I}_k$ to annihilate the first subdiagonal. We follow [4], imposing the matrices $Q_{k+1}(\delta)$ to have determinant 1.

Definition 2.1. *Let $Q_1 = [1]$, and define for $k \geq 1$,*

$$Q_{k+1}(\delta)^* = \Omega_{k+1}(\delta) \begin{bmatrix} Q_k(\delta)^* & 0 \\ 0 & 1 \end{bmatrix}, \quad \Omega_{k+1}(\delta) = \begin{bmatrix} I_{k-1} & 0 & 0 \\ 0 & \overline{c_k(\delta)} & s_k(\delta) \\ 0 & -s_k(\delta) & c_k(\delta) \end{bmatrix},$$

with $s_k(\delta) \geq 0$ and $s_k(\delta)^2 + |c_k(\delta)|^2 = 1$, such that (2.7) *holds.*

Proposition 2.2. *Let $\delta \in \mathbb{C}$, and*

$$\sigma_k(z) := \sqrt{\sum_{j=0}^{k} |q_j(z)|^2}.$$

Then the unitary factor $Q_{k+1}(\delta)^*$ is given by

$$\begin{bmatrix}
-\frac{q_0(\delta)\overline{q_1(\delta)}}{\sigma_1(\delta)\sigma_0(\delta)} & \frac{\sigma_0(\delta)}{\sigma_1(\delta)} & 0 & \cdots & \cdots & 0 \\
-\frac{q_0(\delta)\overline{q_2(\delta)}}{\sigma_2(\delta)\sigma_1(\delta)} & -\frac{q_1(\delta)\overline{q_2(\delta)}}{\sigma_2(\delta)\sigma_1(\delta)} & \frac{\sigma_1(\delta)}{\sigma_2(\delta)} & 0 & & \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & & \ddots & \ddots & 0 \\
-\frac{q_0(\delta)\overline{q_k(\delta)}}{\sigma_k(\delta)\sigma_{k-1}(\delta)} & -\frac{q_1(\delta)\overline{q_k(\delta)}}{\sigma_k(\delta)\sigma_{k-1}(\delta)} & \cdots & \cdots & -\frac{q_{k-1}(\delta)\overline{q_k(\delta)}}{\sigma_k(\delta)\sigma_{k-1}(\delta)} & \frac{\sigma_{k-1}(\delta)}{\sigma_k(\delta)} \\
(-1)^k \frac{q_0(\delta)}{\sigma_k(\delta)} & (-1)^k \frac{q_1(\delta)}{\sigma_k(\delta)} & \cdots & \cdots & (-1)^k \frac{q_{k-1}(\delta)}{\sigma_k(\delta)} & (-1)^k \frac{q_k(\delta)}{\sigma_k(\delta)}
\end{bmatrix}.$$

Moreover,

$$s_k(\delta) = \frac{\sigma_{k-1}(\delta)}{\sigma_k(\delta)} \quad \textit{and} \quad c_k(\delta) = (-1)^k \frac{q_k(\delta)}{\sigma_k(\delta)}, \quad \textit{for } k \geq 1. \tag{2.8}$$

Proof. We leave it to the reader to check that the candidate for $Q_{k+1}(\delta)^*$ indeed has orthonormal rows. It remains to check the subdiagonal and diagonal entries of $Q_{k+1}(\delta)^*(\underline{H}_k - \delta \underline{I}_k)$.

For $k \geq j > \ell$ we find that

$$\begin{aligned}
&\mathbf{e}_j^* Q_{k+1}(\delta)^* (\underline{H}_k - \delta \underline{I}_k)\mathbf{e}_\ell \\
&= -\frac{\overline{q_j(\delta)}}{\sigma_j(\delta)\sigma_{j-1}(\delta)} (q_0(\delta), \dots, q_{j-1}(\delta), \underbrace{0, \dots, 0}_{k+1-j})(\underline{H}_k - \delta \underline{I}_k)\mathbf{e}_\ell \\
&+ \left(\underbrace{0, \dots, 0}_{j}, \frac{\sigma_{j-1}(\delta)}{\sigma_j(\delta)}, \underbrace{0, \dots, 0}_{k-j} \right) (\underline{H}_k - \delta \underline{I}_k)\mathbf{e}_\ell \\
&= -\frac{\overline{q_j(\delta)}}{\sigma_j(\delta)\sigma_{j-1}(\delta)} \left(q_0(\delta), \dots, q_\ell(\delta)\right) (\underline{H}_\ell - \delta \underline{I}_\ell)\mathbf{e}_\ell = 0,
\end{aligned}$$

where the second equality is because of the fact that only the first $\ell + 1 \leq j$ entries of $(\underline{H}_k - \delta \underline{I}_k)\mathbf{e}_\ell$ are nonzero and the third equality because of (2.5). Similarly, for $k \geq j = \ell$,

$$\begin{aligned}
&\mathbf{e}_\ell^* Q_{k+1}(\delta)^* (\underline{H}_k - \delta \underline{I}_k)\mathbf{e}_\ell \\
&= -\frac{\overline{q_\ell(\delta)}}{\sigma_\ell(\delta)\sigma_{\ell-1}(\delta)} (q_0(\delta), \dots, q_{\ell-1}(\delta), \underbrace{0, \dots, 0}_{k+1-\ell})(\underline{H}_k - \delta \underline{I}_k)\mathbf{e}_\ell \\
&+ \left(\underbrace{0, \dots, 0}_{\ell}, \frac{\sigma_{\ell-1}(\delta)}{\sigma_\ell(\delta)}, \underbrace{0, \dots, 0}_{k-\ell} \right) (\underline{H}_k - \delta \underline{I}_k)\mathbf{e}_\ell \\
&= -\frac{\overline{q_\ell(\delta)}}{\sigma_\ell(\delta)\sigma_{\ell-1}(\delta)} (q_0(\delta), \dots, q_{\ell-1}(\delta))(H_\ell - \delta I_\ell)\mathbf{e}_\ell + \frac{\sigma_{\ell-1}(\delta)}{\sigma_\ell(\delta)} h_{\ell+1,\ell} \\
&= -\frac{\overline{q_\ell(\delta)}}{\sigma_\ell(\delta)\sigma_{\ell-1}(\delta)} \left(-q_\ell(\delta) h_{\ell+1,\ell}\right) + \frac{\sigma_{\ell-1}(\delta)}{\sigma_\ell(\delta)} h_{\ell+1,\ell} \\
&= \frac{h_{\ell+1,\ell}}{\sigma_\ell(\delta)\sigma_{\ell-1}(\delta)} \left(|q_\ell(\delta)|^2 + \sigma_{\ell-1}(\delta)^2 \right) = h_{\ell+1,\ell} \frac{\sigma_\ell(\delta)}{\sigma_{\ell-1}(\delta)} > 0.
\end{aligned}$$

Finally, for $k + 1 = j \geq \ell$,

$$\begin{aligned}
&\mathbf{e}_{k+1}^* Q_{k+1}(\delta)^* (\underline{H}_k - \delta \underline{I}_k)\mathbf{e}_\ell \\
&= \frac{(-1)^k}{\sigma_k(\delta)} \left(q_0(\delta), q_1(\delta), \dots, q_k(\delta)\right) \left(\underline{H}_k - \delta I_k\right) \mathbf{e}_\ell = 0,
\end{aligned}$$

according to (2.5). To prove (2.8), observe that

$$Q_{k+1}(\delta)^* = \Omega_{k+1}(\delta) \begin{bmatrix} Q_k(\delta)^* & 0 \\ 0 & 1 \end{bmatrix},$$

if and only if by multiplying on the left with $\begin{bmatrix} \overline{c_k(\delta)} & s_k(\delta) \\ -s_k(\delta) & c_k(\delta) \end{bmatrix}$ we transform

$$\begin{bmatrix} (-1)^{k-1} \frac{q_0(\delta)}{\sigma_{k-1}(\delta)} & (-1)^{k-1} \frac{q_1(\delta)}{\sigma_{k-1}(\delta)} & \cdots & (-1)^{k-1} \frac{q_{k-1}(\delta)}{\sigma_{k-1}(\delta)} & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

into
$$\begin{bmatrix} -\frac{q_0(\delta)\overline{q_k(\delta)}}{\sigma_k(\delta)\sigma_{k-1}(\delta)} & -\frac{q_1(\delta)\overline{q_k(\delta)}}{\sigma_k(\delta)\sigma_{k-1}(\delta)} & \cdots & -\frac{q_{k-1}(\delta)\overline{q_k(\delta)}}{\sigma_k(\delta)\sigma_{k-1}(\delta)} & \frac{\sigma_{k-1}(\delta)}{\sigma_k(\delta)} \\ (-1)^k \frac{q_0(\delta)}{\sigma_k(\delta)} & (-1)^k \frac{q_1(\delta)}{\sigma_k(\delta)} & \cdots & (-1)^k \frac{q_{k-1}(\delta)}{\sigma_k(\delta)} & (-1)^k \frac{q_k(\delta)}{\sigma_k(\delta)} \end{bmatrix},$$

the latter being true for $c_k(\delta)$ and $s_k(\delta)$ as in (2.8). □

Notice that, according to the nested structure of the Hessenberg matrices, $Q_{k+1}(\delta)^*(H_{k+1} - \delta I_{k+1})$ is also upper triangular, but its last diagonal entry given by

$$(-1)^{k+1} h_{k+2,k+1}\, q_{k+1}(\delta)/\sigma_k(\delta)$$

is not necessarily a positive real number. Hence, for obtaining the unitary factor in the unique QR-decomposition of $H_{k+1} - \delta I_{k+1}$ we should rescale the last row of $Q_{k+1}(\delta)^*$ as given in Proposition 2.2 by a phase of modulus 1.

Let us consider the special case of $\delta = 0$ and unitary A as a running example.

Example 2.3. *Suppose that $\delta = 0$ and A is unitary. Then $AV_k = V_{k+1}\underline{H}_k$ and thus $\underline{H}_k$ has orthonormal columns, showing that $\underline{H}_k = Q_{k+1}(0)\underline{I}_k$ and $R_k(0) = I_k$. From Proposition 2.2 we get explicit formulas for the entries of $\underline{H}_k$, in particular, for unitary A,*

$$h_{k+1,k} = \frac{\sigma_{k-1}(0)}{\sigma_k(0)} = s_k(0), \quad (-1)^{k-1}\sigma_{k-1}(0)h_{1,k} = (-1)^k \frac{q_k(0)}{\sigma_k(0)} = c_k(0). \tag{2.9}$$

Furthermore, $AV_k = V_{k+1}Q_{k+1}(0)\underline{I}_k$. Therefore,

$$\begin{aligned} A\mathbf{v}_k &= V_{k+1}\begin{bmatrix} Q_k(0) & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} I_{k-1} & 0 \\ 0 & c_k(0) \\ 0 & s_k(0) \end{bmatrix}\mathbf{e}_k \\ &= V_{k+1}\begin{bmatrix} c_k(0)Q_k(0)\mathbf{e}_k \\ s_k(0) \end{bmatrix} \\ &= s_k(0)\mathbf{v}_{k+1} + (-1)^{k-1}c_k(0)\tilde{\mathbf{v}}_k, \end{aligned}$$

with $\tilde{\mathbf{v}}_k := \frac{1}{\sigma_{k-1}(0)}\sum_{j=0}^{k-1}\overline{q_j(0)}\mathbf{v}_{j+1}$. Also, $\tilde{\mathbf{v}}_{k+1} = s_k(0)\tilde{\mathbf{v}}_k + (-1)^k\overline{c_k(0)}\mathbf{v}_{k+1}$. Hence, the orthonormal vectors $\mathbf{v}_k$ can be constructed using two short recurrence relations:

```
$\tilde{\mathbf{v}}_1 := \mathbf{v}_1$;
for $k = 1, \ldots, N-1$ do
    $\mathbf{z} := A\mathbf{v}_k$;
    $\gamma_k := -\tilde{\mathbf{v}}_k^*\mathbf{z}$;
    $\sigma_k := (1 - |\gamma_k|^2)^{1/2}$;
    $\mathbf{v}_{k+1} := \sigma_k^{-1}(\mathbf{z} + \gamma_k\tilde{\mathbf{v}}_k)$;
    $\tilde{\mathbf{v}}_{k+1} := \sigma_k\tilde{\mathbf{v}}_k + \overline{\gamma}_k\mathbf{v}_{k+1}$;
end
```

Note that $\gamma_k = (-1)^k c_k(0)$ and $\sigma_k = s_k(0)$. The above double recurrence relation is known as the 'Isometric Arnoldi algorithm' designed by Gragg [10].

As a consequence of Proposition 2.2, according to Definition 1.1, we can derive some statements on the rank structure of the unitary Hessenberg matrix $Q_{k+1}(\delta)$, and on a decay property of its entries.

Corollary 2.4. *$Q_{k+1}(\delta)$ is $(0,1)$-upper-separable*[1]*. Moreover, for the submatrix $\widetilde{Q}$ of $Q_{k+1}(\delta)$ formed with the first $m \le k+1$ rows and the last $\ell \le k-m+2$ columns we have that*

$$\|\widetilde{Q}\| = \sigma_{m-1}(\delta)/\sigma_{k-\ell+1}(\delta).$$

Proof. The first statement follows by observing that

$$Q_{k+1}(\delta) = \begin{bmatrix} \overline{q_0(\delta)} \\ \vdots \\ \overline{q_k(\delta)} \end{bmatrix}\mathbf{e}_1^* Q_{k+1}(\delta) + L_{k+1}(\delta), \tag{2.10}$$

with $L_{k+1}(\delta)$ strictly lower triangular, i.e., with zero entries on the main diagonal. This is a direct consequence of Proposition 2.2. In particular, we deduce that $\widetilde{Q}$ is of rank 1. From Proposition 2.2 it follows that

$$\begin{aligned} \|\widetilde{Q}\|^2 &= \frac{\sigma_{m-1}(\delta)^2}{\sigma_k(\delta)^2} + \sum_{j=k-\ell+2}^{k}\frac{|q_j(\delta)|^2}{\sigma_j(\delta)^2\sigma_{j-1}(\delta)^2}\sigma_{m-1}(\delta)^2 \\ &= \frac{\sigma_{m-1}(\delta)^2}{\sigma_k(\delta)^2} + \sum_{j=k-\ell+2}^{k}\frac{\sigma_j(\delta)^2 - \sigma_{j-1}(\delta)^2}{\sigma_j(\delta)^2\sigma_{j-1}(\delta)^2}\sigma_{m-1}(\delta)^2 \\ &= \sigma_{m-1}(\delta)^2\left(\frac{1}{\sigma_k(\delta)^2} + \sum_{j=k-\ell+2}^{k}\left(\frac{1}{\sigma_{j-1}(\delta)^2} - \frac{1}{\sigma_j(\delta)^2}\right)\right), \end{aligned}$$

[1] This implies that $Q_{k+1}(\delta)$ is quasi separable, but in general not semiseparable.

giving the claimed result. □

We end this subsection by observing that Corollary 2.4 immediately implies a rank property as well as a decay of entries for resolvents of H_{k+1}.

Corollary 2.5. *Suppose that $H_{k+1} - \delta I_{k+1}$ is invertible. Then $(H_{k+1} - \delta I_{k+1})^{-*}$ is $(0,1)$-upper-separable.*

Moreover, for the submatrix $\widetilde{H}$ of $(H_{k+1} - \delta I_{k+1})^{-*}$ formed with the first $m \leq k+1$ rows and the last $\ell \leq k-m+2$ columns we have that

$$\|\widetilde{H}\| \leq \|(H_{k+1} - \delta I_{k+1})^{-1}\| \, \sigma_{m-1}(\delta)/\sigma_{k-\ell+1}(\delta).$$

Proof. Let us write $H_{k+1} - \delta I_{k+1} = Q_{k+1}(\delta) R$ with upper triangular and invertible R. Then

$$(H_{k+1} - \delta I_{k+1})^{-*} = Q_{k+1}(\delta) R^{-*}$$

with R^{-*} lower triangular of norm $\|(H_{k+1} - \delta I_{k+1})^{-1}\|$. Replacing $Q_{k+1}(\delta)$ by (2.10) yields

$$(H_{k+1} - \delta I_{k+1})^{-*} = (q_0(\delta), \ldots, q_k(\delta))^* \, \mathbf{e}_1^* (H_{k+1} - \delta I_{k+1})^{-*} + \widetilde{L}_{k+1}(\delta), \tag{2.11}$$

with $\widetilde{L}_{k+1}(\delta)$ strictly lower triangular; proving the first statement. This implies that

$$\widetilde{H} = (q_0(\delta), ..., q_{m-1}(\delta))^* \, \mathbf{e}_1^* \widetilde{H},$$

and thus $\|\widetilde{H}\| = \sigma_{m-1}(\delta) \, \|\mathbf{e}_1^* \widetilde{H}\|$. Notice that $\mathbf{e}_1^* \widetilde{H}$ is obtained by multiplying the first row of $Q_{k+1}(\delta)$ with the last ℓ columns of R^{-*}. As R^{-*} is lower triangular, $\mathbf{e}_1^* \widetilde{H} = \widetilde{Q}\widetilde{R}$, $\widetilde{Q}$ a row vector formed with the last ℓ entries of the first row of $Q_{k+1}(\delta)$ and $\widetilde{R}$ the lower-right $\ell \times \ell$ minor of R^{-*}. Therefore, applying Corollary 2.4 to $\widetilde{Q}$ yields

$$\|\mathbf{e}_1^* \widetilde{H}\| \leq \|\widetilde{Q}\| \, \|\widetilde{R}\| \leq \frac{\sigma_0(\delta)}{\sigma_{k-\ell+1}(\delta)} \|R^{-*}\|,$$

which together with $\|\widetilde{H}\| = \sigma_{m-1}(\delta) \, \|\mathbf{e}_1^* \widetilde{H}\|$ and $\|R^{-*}\| = \|(H_{k+1} - \delta I_{k+1})^{-1}\|$ proves the second statement. □

2.3 The GMRES residual and orthogonal polynomials

We will give some more details on the link between the GMRES residual vectors and orthogonal polynomials. More specifically, we will write the GMRES residual vector as a linear combination of Arnoldi vectors. In the particular case of unitary A, an even nicer relation arises for the GMRES residual vectors.

In what follows we denote by $\mathbf{r}_k(\delta)$ the kth GMRES residual for the shifted system $(A - \delta I_n)\mathbf{x} = \mathbf{b}$, with starting vector $\mathbf{x}_0 = 0$, and denote by $\mathbf{w}_k(\delta) := \mathbf{r}_k(\delta)/\|\mathbf{r}_k(\delta)\|$ its normalized version.

Proposition 2.6. *The kth GMRES residual for the shifted system $(A - \delta I_n)\mathbf{x} = \mathbf{b}$ with starting vector $\mathbf{x}_0 = 0$, can be written as*

$$\mathbf{r}_k(\delta) = p_k(A)\mathbf{r}_0(\delta), \quad \textit{where} \quad p_k(z) = \frac{1}{\sigma_k^2(\delta)} \sum_{j=0}^{k} \overline{q_j(\delta)} q_j(z). \tag{2.12}$$

For its normalized version we have

$$\mathbf{w}_k(\delta) = \frac{\mathbf{r}_k(\delta)}{\|\mathbf{r}_k(\delta)\|} = \frac{1}{\sigma_k(\delta)} \sum_{j=0}^{k} \overline{q_j(\delta)} \mathbf{v}_{j+1}. \tag{2.13}$$

Proof. Since we choose as starting vector $\mathbf{x}_0 = 0$, we find the initial GMRES residual $\mathbf{r}_0(\delta) = \mathbf{b} - (A - \delta I_n)0 = \mathbf{b} = \mathbf{v}_1 \, \|\mathbf{b}\|$. Then we have

$$\mathbf{r}_k(\delta) = \mathbf{r}_0(\delta) - (A - \delta I_n) V_k \mathbf{y}, \text{ with } \mathbf{y} = \operatorname{argmin}_{\mathbf{y}} \|\mathbf{r}_0(\delta) - (A - \delta I_n) V_k \mathbf{y}\|. \tag{2.14}$$

Note that $\mathbf{r}_0 - (A - \delta I_n)V_k\mathbf{y}$ can be written as $p_k(A)\mathbf{r}_0(\delta)$ with p_k a polynomial of degree at most k and $p_k(\delta) = 1$. Then $\mathbf{r}_k(\delta) = p_k(A)\mathbf{r}_0$, $p_k = \widetilde{p_k}/\widetilde{p_k}(\delta)$, with

$$\widetilde{p_k} = \mathrm{argmin}_{p\in\mathcal{P}_k} \frac{\|p(A)\mathbf{r}_0(\delta)\|}{|p(\delta)|} = \mathrm{argmin}_{p\in\mathcal{P}_k} \frac{\|p\|_{A,\mathbf{v}_1}}{|p(\delta)|}, \tag{2.15}$$

where $\mathcal{P}_k$ denotes the set of polynomials of degree at most k, and we use the norm induced by (2.2). Note that if $p(z) = \sum_{j=0}^k c_j q_j(z)$, then $\|p\|_{A,\mathbf{v}_1}^2 = \sum_{j=0}^k |c_k|^2$ by orthonormality. Therefore, the Cauchy-Schwarz inequality yields

$$\frac{\|p\|_{A,\mathbf{v}_1}}{|p(\delta)|} \geq \frac{1}{\sigma_k(\delta)}, \tag{2.16}$$

where the minimum is attained for $c_j = \overline{q_j(\delta)}$. Combining (2.16) and (2.15) we conclude that (2.12) holds. In particular, $\|\mathbf{r}_k(\delta)\| = \|b\|/\sigma_k(\delta)$, which together with (2.6) implies (2.13). □

Remark 2.7. *As $\sigma_{m-1}(\delta)/\sigma_{N-\ell+1} = \|\mathbf{r}_{N-\ell+1}(\delta)\|/\|\mathbf{r}_{m-1}(\delta)\|$, we see that the decay rates in Corollary 2.4 and Corollary 2.5 are linked to the convergence of the GMRES algorithm. If the GMRES algorithm converges faster, the decay pattern becomes more pronounced.*

Remark 2.8. *By taking norms in (2.12), we see that for the relative GMRES residual*

$$\frac{\|\mathbf{r}_k(\delta)\|}{\|\mathbf{r}_0(\delta)\|} = \frac{1}{\sigma_k(\delta)} = \min_{p\in\mathcal{P}_k} \frac{\|p(A)\mathbf{b}\|}{\|b\||p(\delta)|}.$$

More generally, for the decay rates in Corollary 2.4 and Corollary 2.5 for $\ell \leq k - m + 2$, we have

$$\frac{\sigma_{m-1}(\delta)}{\sigma_{k-\ell+1}(\delta)} \leq \min_{p\in\mathcal{P}_{k+2-m-\ell}} \frac{\|p(A)\mathbf{w}_{m-1}(\delta)\|}{\|\mathbf{w}_{m-1}(\delta)\|\,|p(\delta)|}, \tag{2.17}$$

Let $\Omega \subset \mathbb{C}$ be a simply connected and compact K-spectral set for A; that is, $\|\pi(A)\| \leq K \max_{z\in\Omega} |\pi(z)|$ for all polynomials π (and hence $\Lambda(A) \subset \Omega$) with $\delta \notin \Omega$. Also, let φ be a map, mapping $\mathbb{C} \setminus \Omega$ conformally onto the exterior of the closed unit disk. Then it can be shown that the right-hand side of (2.17) can be bounded above by $|1/\varphi(\delta)|^{k+2-\ell-m} < 1$ times a modest constant, see, e.g., [16, Chapter 6.11.2] for the case where Ω is an ellipse.

Example 2.9. *As in Example 2.3, assume the matrix A is unitary. Given a polynomial p of degree k, its reversed polynomial p^* is defined as*

$$p^*(z) := z^k\overline{p(1/\overline{z})}.$$

The orthogonal polynomials $q_k(z)$ can be expressed as

$$\begin{aligned} q_k(z) &= q_k^*(0)\ \mathit{argmin}_{q\ \mathit{monic\ of\ degree}\ k} \|q\|_{A,\mathbf{v}_1} \\ &= q_k^*(0)\ \mathit{argmin}_{q\ \mathit{of\ degree}\ k} \frac{\|q\|_{A,\mathbf{v}_1}}{|q_k^*(0)|}. \end{aligned} \tag{2.18}$$

Note that $q_k^*(0)$ is the leading coefficient of q_k. As A is unitary, $\|q\|_{A,\mathbf{v}_1} = \|q^*\|_{A,\mathbf{v}_1}$. Hence, (2.18) can be rewritten as

$$q_k(z) = q_k^*(0)\mathit{argmin}_{q\ \mathit{of\ degree}\ k} \frac{\|q^*\|_{A,\mathbf{v}_1}}{|q^*(0)|}$$

Therefore,

$$q_k^*(z) = q_k^*(0)\mathit{argmin}_{q\in\mathcal{P}_k, q(0)\neq 0} \frac{\|q\|_{A,\mathbf{v}_1}}{|q(0)|}. \tag{2.19}$$

Because of (2.4), $q_k^*(0) \geq 0$. Also, $\|q_k^*\|_{A,\mathbf{v}_1} = 1$. Combining this with (2.15) and (2.12), (2.19) yields

$$q_k^*(z) = \frac{1}{\sigma_k(0)} \sum_{j=0}^k \overline{q_j(0)} q_j(z), \tag{2.20}$$

Therefore, the kth normalized GMRES residual $\mathbf{w}_k(0)$ of a unitary matrix A can be expressed as

$$\mathbf{w}_k(0) = q_k^*(A)\mathbf{v}_1. \tag{2.21}$$

2.4 A progressive GMRES residual formula

By (2.13) we have that the kth normalized GMRES residual satisfies

$$\begin{aligned}
\mathbf{w}_k(\delta) &= \frac{1}{\sigma_k(\delta)}\sum_{i=0}^{k}\overline{q_i(\delta)}\mathbf{v}_{i+1} \\
&= \frac{\sigma_{k-1}(\delta)}{\sigma_k(\delta)}\frac{1}{\sigma_{k-1}(\delta)}\sum_{i=0}^{k-1}\overline{q_i(\delta)}\mathbf{v}_{i+1} + \frac{\overline{q_k(\delta)}}{\sigma_k(\delta)}\mathbf{v}_{k+1} \\
&= \frac{\sigma_{k-1}(\delta)}{\sigma_k(\delta)}\mathbf{w}_{k-1}(\delta) + \frac{\overline{q_k(\delta)}}{\sigma_k(\delta)}\mathbf{v}_{k+1}, \\
&= s_k(\delta)\mathbf{w}_{k-1}(\delta) + (-1)^k\overline{c_k(\delta)}\mathbf{v}_{k+1}, \qquad (2.22)
\end{aligned}$$

the last equality following from (2.8). This demonstrates the existence of an updating formula to compute the residual vectors progressively. This formula can also be derived by means of the QR-factorization of $\underline{H}_k - \delta\underline{I}_k$, see Proposition 2.2 and [16, Subsection 6.5.3]. The next result shows that it is not necessary to compute such a factorization for obtaining $s_k(\delta)$ and $c_k(\delta)$, if one is willing to compute the additional scalar product (2.23).

Proposition 2.10. *Define* $\tau_k(\delta) = \mathbf{e}_k^* Q_k(\delta)^*(H_k - \delta I_k)\mathbf{e}_k$. *Then*

$$\tau_k(\delta) = \mathbf{w}_{k-1}(\delta)^*(A - \delta I_n)\mathbf{v}_k(-1)^{k-1}, \qquad (2.23)$$

and

$$s_k(\delta) = \frac{h_{k+1,k}}{\sqrt{h_{k+1,k}^2 + |\tau_k(\delta)|^2}} \quad \textit{and} \quad c_k(\delta) = \frac{\tau_k(\delta)}{\sqrt{h_{k+1,k}^2 + |\tau_k(\delta)|^2}}. \qquad (2.24)$$

Proof. From Proposition 2.2 and (2.13) it follows that

$$\begin{aligned}
V_k Q_k(\delta)\mathbf{e}_k &= V_k \begin{pmatrix} \overline{q_0(\delta)} \\ \vdots \\ \overline{q_{k-1}(\delta)} \end{pmatrix} \frac{(-1)^{k-1}}{\sigma_{k-1}(\delta)} \\
&= \frac{(-1)^{k-1}}{\sigma_{k-1}(\delta)}\sum_{i=1}^{k}\mathbf{v}_i\overline{q_{i-1}(\delta)} = (-1)^{k-1}\mathbf{w}_{k-1}(\delta). \qquad (2.25)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\tau_k(\delta) &= \mathbf{e}_k^* Q_k(\delta)^*(H_k - \delta I_k)\mathbf{e}_k \\
&= \mathbf{e}_k^* Q_k(\delta)^* V_k^* V_k(H_k - \delta I_k)\mathbf{e}_k \\
&= \mathbf{w}_{k-1}(\delta)^*(A\mathbf{v}_k - h_{k+1,k}\mathbf{v}_{k+1} - \delta\mathbf{v}_k)(-1)^{k-1} \\
&= \mathbf{w}_{k-1}(\delta)^*(A - \delta I_n)\mathbf{v}_k(-1)^{k-1},
\end{aligned}$$

establishing (2.23). We now claim that

$$\mathbf{w}_k(\delta) \perp (A - \delta I_n)\mathbf{v}_j, \quad \text{for } 1 \le j \le k, \qquad (2.26)$$

which is a direct consequence of (2.14):

$$\mathbf{r}_k(\delta) = \mathbf{r}_0(\delta) - \mathbf{u}_k \quad \text{with} \quad \mathbf{u}_k = \text{argmin}_{\mathbf{u}\in\mathcal{K}}\|\mathbf{r}_0(\delta) - \mathbf{u}\|,$$

where $\mathcal{K} = (A - \delta I_n)\text{span}\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$.

It remains to prove (2.24). Taking inner products with $(A - \delta I_n)\mathbf{v}_k$ in all terms of (2.22) and making use of (2.23) and (2.26), results in

$$0 = s_k(\delta)\overline{\tau_k(\delta)}(-1)^{k-1} + \overline{c_k(\delta)}h_{k+1,k}(-1)^k. \qquad (2.27)$$

Together with $s_k^2(\delta) + |c_k(\delta)|^2 = 1$, this yields (2.24). □

Example 2.11. *Let us return to the particular case of $\delta = 0$ and unitary A as discussed in Examples 2.3 and 2.9. Inserting* (2.6) *and* (2.21) *in* (2.22) *and identifying the underlying polynomials gives*

$$q_k^*(z) = s_k(0)q_{k-1}^*(z) + (-1)^k\overline{c_k(0)}q_k(z). \tag{2.28}$$

Since $q_{k-1}^*(z) - q_{k-1}^*(0)$ is z times a polynomial of degree $< k-1$ and $q_0(z) = 1$, we get from (2.23) *that*

$$\begin{aligned}\tau_k(0) &= (-1)^{k-1}\langle zq_{k-1}, q_{k-1}^*\rangle_{A,\mathbf{v}_1} \\ &= (-1)^{k-1}\overline{q_{k-1}^*(0)}\langle zq_{k-1}, q_0\rangle_{A,\mathbf{v}_1} \\ &= (-1)^{k-1}\sigma_{k-1}(0)h_{1,k} = c_k(0)\end{aligned}$$

and $h_{k+1,k} = s_k(0)$, where we applied (2.20) *and* (2.9)*. Notice that this simplification for unitary A is in accordance with* (2.24)*. Taking the star operation in* (2.28) *gives the second relation*

$$q_k(z) = s_k(0)zq_{k-1}(z) + (-1)^k c_k(0)q_k^*(z). \tag{2.29}$$

We should mention that, in case of unitary A, the scalar product (2.2) *can be written as a scalar product in terms of a (discrete) measure supported on the unit circle. Thus we have the whole theory of orthogonal polynomials on the unit circle, and in particular relations* (2.28) *and* (2.29) *are known as the Szegő recurrence relations, a coupled two-term recurrence for orthonormal polynomials on the unit circle [11, Formula 1.2-1.7].*

3 The Arnoldi process for BML-matrices

We will establish a fast variant of the Arnoldi process which is applicable to the class of BML-matrices as described by formula (1.1). In §3.1 we will use the explicit representation of $(H_{k+1} - \delta I_{k+1})^{-*}$ in terms of orthogonal polynomials as stated in (2.11) to demonstrate that the underlying Hessenberg matrix has an upper-separable structure. The assumption on simple poles makes it possible to easily express generators for the upper-separable structure in polynomial language. In particular, the GMRES residual vectors, which can be expressed in terms of orthogonal polynomials by (2.13), are showing up. In §3.2 we will see that the orthonormal basis vectors (up to a correction term incorporating the low rank perturbation in (1.1)) can be written as a linear combination of previously computed basis vectors and GMRES residual vectors. In §3.3 the results from §3.2 and §2.4, enabling to compute the necessary GMRES residual vectors progressively, will be combined into a new Arnoldi iteration for BML-matrices.

3.1 Structure formula for BML-matrices

A rank structure revealing formula for BML-matrices will be derived in terms of the orthogonal polynomials q_k defined in §2.1. This formula (3.3) will be the key for the design of short recurrence relations in the following subsection. We first show that the BML-structure (1.1) is inherited by the underlying Hessenberg matrix.

Proposition 3.1. *Let A be an invertible matrix satisfying relation* (1.1)*, and denote by $N \le n$ the Arnoldi termination index. Then H_N is also a BML-matrix with the same polynomials p, q and indices $\widetilde{m}_1 = m_1$, $\widetilde{m}_2 = m_2$ and $\widetilde{m}_3 \le m_3$. More precisely,*

$$H_N^* = p(H_N)q(H_N)^{-1} + F_N G_N^* \tag{3.1}$$

where $F_k := V_k^* F$ and $G_k := V_k^* G$.

Proof. By definition of N we have $AV_N = V_N H_N$, i.e., the columns of V_N span an invariant subspace of A. By recurrence on the degree one shows that $\pi(A)V_N = V_N\pi(H_N)$, or $\pi(H_N) = V_N^*\pi(A)V_N$, for any polynomial π. Moreover, if $\pi(A)$ is of full rank, then so is $\pi(H_N)$. In particular, since we assume $q(A)$ to be invertible, so is the matrix $q(H_N)$. Hence, $V_N q(H_N)^{-1} = q(A)^{-1}V_N$. This implies that

$$\begin{aligned}H_N^* - p(H_N)q(H_N)^{-1} &= V_N^*(A^* - p(A)q(A)^{-1})V_N \\ &= V_N^* F G^* V_N,\end{aligned}$$

as claimed in (3.1). □

A combination with Corollary 2.5 gives us the following result.

Corollary 3.2. *Let A be a matrix satisfying relation* (1.1)*. Denote by $N \leq n$ the Arnoldi termination index, and $m := \max(0, m_1 - m_2 + 1)$. Then $H_N \in \mathbb{C}^{N\times N}$ is $(m, m_2+m_3)-$upper-separable. More precisely,*

$$((H_N)_{k,\ell})_{\ell=k+m,\dots,N} = \sum_{j=1}^{m_2} \overline{d_j q_{k-1}(z_j)} \left(((H_N - z_j I_N)^{-*})_{1,\ell} \right)_{\ell=k+m,\dots,N} + \left((G_N F_N^*)_{k,\ell} \right)_{\ell=k+m,\dots,N}, \tag{3.2}$$

where $z_1, \dots, z_{m_2}$ denote the poles of the rational function $p(z)/q(z)$.

Proof. The fact that H_N is $(m, m_2+m_3)-$upper-separable is already known from [14]. Below we give an alternative, more constructive, proof leading to explicit generators for the low rank part of H_N. By assumption, the rational function p/q in (1.1) has simple poles and thus has the partial fraction decomposition

$$\frac{p(z)}{q(z)} = \sum_{j=1}^{m_2} \frac{d_j}{z - z_j} + \pi(z),$$

for some constants z_j, d_j and a polynomial π of degree $m_1 - m_2$ ($\pi = 0$ if $m_1 < m_2$). Replacing z by H_N, taking adjoints and using (3.1) and (2.11) leads to

$$\begin{aligned} & H_N - G_N F_N^* - \pi(H_N)^* \\ = \; & \sum_{j=1}^{m_2} \overline{d_j} (H_N - z_j I_N)^{-*} \\ = \; & L_N + \sum_{j=1}^{m_2} \overline{d_j} \Big(q_0(z_j), \dots, q_{N-1}(z_j) \Big)^* \mathbf{e}_1^* (H_N - z_j I_N)^{-*}, \end{aligned} \tag{3.3}$$

with a strictly lower triangular matrix L_N. Finally, according to the upper Hessenberg structure of H_N, the matrix $\pi(H_N)^*$ has zero entries on and above the mth diagonal, establishing the upper-separable structure and formula (3.2). □

Remark 3.3. *The assumption on simple poles is not necessary for H_N to have an upper-separable structure. However, once we drop this constraint, it is not clear whether or not there exists a link between the generators of the low-rank structure and the GMRES residual vectors. We refer to [14] for more information on this topic.*

Note that H_k for $k < N$ also has an upper-separable structure as it is a leading principal minor (submatrix) of H_N. However, H_k does not satisfy the same matrix equation as H_N.

3.2 Short recurrence relations for BML-matrices

We will now derive a short recurrence relation for BML-matrices. To do so, the vector

$$\mathbf{v}_k' := A\mathbf{v}_k - V_{k-m} G_{k-m} F^* \mathbf{v}_k, \tag{3.4}$$

is introduced, which is equal to $A\mathbf{v}_k$ up to a correction term induced by the low rank perturbation in (1.1). In [4] it is shown that for nearly Hermitian A, i.e., $p(z) = z$ and $q(z) = 1$ in (1.1), the Arnoldi vectors satisfy a three term recurrence relation. More specifically,

$$\mathbf{v}_k' = h_{k-1,k}\mathbf{v}_{k-1} + h_{k,k}\mathbf{v}_k + h_{k+1}\mathbf{v}_{k+1,k}, \tag{3.5}$$

where $h_{k-1,k} = \mathbf{v}_{k-1}^* \mathbf{v}_k'$, $h_{k,k} = \mathbf{v}_k^* \mathbf{v}_k'$ and $h_{k+1,k} = \mathbf{v}_{k+1}^* \mathbf{v}_k'$ are entries of the underlying Hessenberg matrix. We refer to [4] for a detailed discussion. For a general BML-matrix, Proposition 3.4 states that $\mathbf{v}_k'$ is a linear combination of GMRES residual vectors and Arnoldi vectors, including $\mathbf{v}_{k+1}$. As we will discuss below, this results in a short recurrence relation which reduces to (3.5) in the specific case of nearly Hermitian matrices and which can be used to compute the Arnoldi vectors in an efficient way.

Proposition 3.4. *Let A be a matrix satisfying relation* (1.1)*, and denote by $N \leq n$ the Arnoldi termination index, and $m := \max(0, m_1 - m_2 + 1)$. Then for $m < k \leq N$, the vector $\mathbf{v}'_k$ as defined by* (3.4) *can be written as a linear combination of $\mathbf{v}_j$ for $j = k-m+1, ..., k+1$ and of $\mathbf{w}_{k-m-1}(z_j)$ for $j = 1, ..., m_2$. More precisely,*

$$\mathbf{v}'_k = \sum_{j=1}^{m_2} a_{j,k}\mathbf{w}_{k-m-1}(z_j) + \sum_{j=k-m+1}^{k+1} h_{j,k}\mathbf{v}_j, \tag{3.6}$$

for some constants $a_{j,k}$, $h_{j,k}$ entries of the underlying Hessenberg matrix H_N, and $m < k \leq N$.

Proof. Notice that by construction,

$$\mathbf{v}''_k := \mathbf{v}'_k - \sum_{j=k-m+1}^{k+1} h_{j,k}\mathbf{v}_j$$

lies in the Krylov space spanned by the columns of V_{k-m}. As a result we have

$$\mathbf{v}''_k = V_{k-m}V^*_{k-m}\mathbf{v}''_k = V_{k-m}V^*_{k-m}\mathbf{v}'_k.$$

Define

$$\widehat{I}_{k-m} := \begin{bmatrix} I_{k-m} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{N\times(k-m)}.$$

Then $V_{k-m} = V_N\widehat{I}_{k-m}$. Combining the above yields

$$\begin{aligned} \mathbf{v}'_k &= \mathbf{v}'_k - \mathbf{v}''_k + V_{k-m}V^*_{k-m}\mathbf{v}'_k \\ &= \mathbf{v}'_k - \mathbf{v}''_k + V_{k-m}\left(V^*_{k-m}AV_N - V^*_{k-m}GF^*_N\right)\mathbf{e}_k \\ &= \mathbf{v}'_k - \mathbf{v}''_k + V_{k-m}\widehat{I}^*_{k-m}\left(H_N - G_NF^*_N\right)\mathbf{e}_k. \end{aligned}$$

The vector $\widehat{I}^*_{k-m}\left(H_N - G_NF^*_N\right)\mathbf{e}_k \in \mathbb{C}^{k-m}$ consists of the first $k-m$ components of the kth column of $H_N - G_NF^*_N$. As the entries of L_N and $\pi(H_N)^*$ in (3.3) are zero on and above the mth diagonal, it follows that

$$\widehat{I}^*_{k-m}\left(H_N - G_NF^*_N\right)\mathbf{e}_k = \sum_{j=1}^{m_2} \overline{d_j}\left(q_0(z_j), ..., q_{k-m-1}(z_j)\right)^* \mathbf{e}_1(H_N - z_jI_N)^{-*}\mathbf{e}_k,$$

which by (2.13) can be rewritten as

$$V_{k-m}\widehat{I}^*_{k-m}\left(H_N - G_NF^*_N\right)\mathbf{e}_k = \sum_{j=1}^{m_2} a_{j,k}\mathbf{w}_{k-m-1}(z_j),$$

with $a_{j,k} := \overline{d_j}\sigma_{k-m-1}(z_j)\mathbf{e}_1(H_N - z_jI_N)^{-*}\mathbf{e}_k$. As a result, $\mathbf{v}'_k$ can be written as $\mathbf{v}'_k - \mathbf{v}''_k$, a linear combination of $\mathbf{v}_{k+1}, ..., \mathbf{v}_{k-m+1}$ (with coefficients being entries of the underlying Hessenberg matrix), plus a linear combination of $\mathbf{w}_{k-m-1}(z_1), ..., \mathbf{w}_{k-m-1}(z_m)$. □

Remark 3.5. *Proposition 3.4 remains true if we replace $\mathbf{w}_{k-m-1}(z_j)$ by $\mathbf{w}_\ell(z_j)$ or/and $V_{k-m}G_{k-m}$ by $V_{\ell+1}G_{\ell+1}$ for $\ell \in \{k-m-1, k-m, ..., k\}$. This is the direct result of the observation that both $V_{k-m}G_{k-m}F^*\mathbf{v}_k - V_{\ell+1}G_{\ell+1}F^*\mathbf{v}_k$ and $\mathbf{w}_{k-m-1}(z_j) - \sigma_\ell(z_j)/\sigma_{k-m-1}(z_j)\mathbf{w}_\ell(z_j)$ are elements of span$(\mathbf{v}_{k-m+1}, \ldots, \mathbf{v}_{\ell+1})$ if $\ell > k-m-1$ and equal to zero if $\ell = k-m-1$. However, choosing $\ell \neq k-m-1$ causes to lose orthogonality between $\mathbf{v}'_k - \mathbf{v}''_k$ and $\mathbf{v}''_k$ and to loose the link with the entries of the underlying Hessenberg matrix in* (3.6)*.*

Next, let us show how the recurrence relation of Proposition 3.4 reduces to the well-known Szegő recurrence if the matrix under consideration is unitary.

Example 3.6. *Let us return to the particular case of $\delta = 0$ and unitary A as discussed in Examples 2.3, 2.9, and 2.11. Inserting* (2.6) *and* (2.21) *in the second Szegő relation* (2.29) *leads to*

$$\mathbf{v}_{k+1} = s_k(0)A\mathbf{v}_k + (-1)^k c_k(0)\mathbf{w}_k(0)$$

which is exactly the variant with $\ell = k, m_2 = m_3 = 1, m_1 = m = 0$ of Proposition 3.4 discussed in Remark 3.5.

Remark 3.7. *In [5] a class of matrices, called (H, m)-well-free matrices is investigated, and it is established that these matrices satisfy the recurrence relation* (1.3) *with $m_1 = m_2, m_3 = 0$. Briefly described, these matrices form a subset of upper-separable Hessenberg matrices which satisfy an additional constraint preventing a breakdown in the recurrence relation* (1.3)*. Intuitively, this additional "well-free" constraint signifies that there are no rank deficiencies encountered in the low rank part of the upper-separable Hessenberg matrix. The problem of breakdown is discussed in [2] and [14], where it is overcome by making use of a set of several multiple recurrence relations instead of the single recurrence relation* (1.3)*, and the use of an algorithm based on* (1.3) *which provides a set of column vectors to generate the low rank structure of the underlying Hessenberg matrix, respectively. We will, however, not need any well-free constraint to prevent a breakdown in the recurrence relation stated in Proposition 3.4, as our approach does not impose any limitations on the matrix structure beyond* (1.1)*.*

3.3 The algorithm

Algorithm 1, which we will name *Fast Arnoldi* throughout the text, describes a fast variant of the Arnoldi algorithm for BML-matrices based on Proposition 3.4. We will give a short description of each of the components of the algorithm and print the corresponding piece of pseudocode.

The idea is to make alternate use of the recurrence relations

$$\begin{aligned} \mathbf{v}'_k &= \sum_{j=1}^{m_2} a_{j,k}\mathbf{w}_{k-m-1}(z_j) + \sum_{j=k-m+1}^{k+1} h_{j,k}\mathbf{v}_j, && (3.7) \\ \mathbf{w}_k(z_j) &= s_k(z_j)\mathbf{w}_{k-1}(z_j) + (-1)^k\overline{c_k(z_j)}\mathbf{v}_{k+1}. && (3.8) \end{aligned}$$

The recurrence relation (3.7) is used to compute the next orthonormal basis vector of the Krylov subspace as a linear combination of previously computed orthonormal vectors as well as GMRES residual vectors, while the recurrence relation (3.8) is used to update the GMRES residual vectors once a new orthonormal vector is retrieved. As (3.7) is only valid for $k > m$, the first m orthonormal vectors are computed by means of the classical Arnoldi iteration.

Each time the relation (3.7) is employed, the vector $\mathbf{v}'_k$ is formed, causing products between vectors and matrices to be computed. The total complexity to compute $\mathbf{v}'_k$ is $\mathcal{O}(m_3 n) + \mathcal{O}(n^2)$.

```
$\hat{F}_k := \mathbf{v}_k^* F$; $\hat{G}_{k-m} := \mathbf{v}_{k-m}^* G$;
$\widetilde{G} := \widetilde{G} + \mathbf{v}_{k-m}\hat{G}_{k-m}$;
$\mathbf{v}' := A\mathbf{v}_k - \widetilde{G}\hat{F}_k^*$;
```

The coefficients $a_{j,k}$ and $h_{j,k}$ in (3.7) are the solution of the least squares problem

$$\mathbf{v}'_k = [\mathbf{w}_{k-m-1}(z_1), \ldots, \mathbf{w}_{k-m-1}(z_{m_2}), \mathbf{v}_{k-m+1}, \ldots, \mathbf{v}_{k+1}] \begin{bmatrix} a_{1,k} \\ \vdots \\ a_{m_2,k} \\ h_{k-m+1,k} \\ \vdots \\ h_{k+1,k} \end{bmatrix}.$$

Note that $\mathbf{v}_i \perp \operatorname{span}\{\mathbf{w}_{k-m-1}(z_1), \ldots, \mathbf{w}_{k-m-1}(z_{m_2})\}$ for all $k-m+1 \le i \le k+1$, allowing to solve the above least squares problem without knowing $\mathbf{v}_{k+1}$ in advance. To shorten notation in subsequent discussions, we define $M_k := [\mathbf{w}_{k-m-1}(z_1), \ldots, \mathbf{w}_{k-m-1}(z_{m_2})]$.

Each of the coefficients $h_{j,k}$ are entries of the kth column of the corresponding Hessenberg matrix and are computed as $\mathbf{v}_j^*\mathbf{v}'_k$, which has a computational complexity of $\mathcal{O}(mn)$.

```
for $j = k, k-1, \ldots, k-m+1$ do
    $h_{j,k} := \mathbf{v}_j^*\mathbf{v}'$; $\mathbf{v}' := \mathbf{v}' - h_{j,k}\mathbf{v}_j$;
end
```

Next, a QR-decomposition of the matrix M_k is computed, after which the coefficients $a_{1,k}, \ldots a_{m_2,k}$ are retrieved by back-substitution. The complexity of this operation is $\mathcal{O}\left(m_2^2(n+1)\right)$.

```
    $Q := [\mathbf{q}_1, \ldots, \mathbf{q}_{m_2}], R := (r_{i,j}) \in \mathbb{C}^{m_2 \times m_2}$,
    such that $QR = M_k$;
    for $j = m_2, m_2 - 1, \ldots, 1$ do
        $a_{j,k} := \left(\mathbf{q}_j^* \mathbf{v}' - \sum_{\ell=j+1}^{m_2} r_{j,\ell} a_{j,k}\right) / r_{j,j}$;
        $\mathbf{v}' = \mathbf{v}' - a_{j,k} \mathbf{w}_{k-m-1}(z_j)$;
    end
    $h_{k+1,k} = ||\mathbf{v}'||$; $\mathbf{v}_{k+1} := \mathbf{v}' / h_{k+1,k}$;
```

Then for each z_i, $1 \leq i \leq m_2$, recurrence relation (3.8) is used to update the GMRES residual vectors, which are all equal to the starting vector $\mathbf{v}_1$ at the beginning of the iteration ($k = m + 1$). Each time the relation (3.8) is employed, a matrix vector product needs to be computed. This leads to a total complexity of $\mathcal{O}(m_2 n^2)$.

```
    for $j = 1, \ldots, m_2$ do
        $\tau_{k-m}(z_j) = (-1)^{k-m-1} \mathbf{w}_{k-m-1}(z_j)^* (A - z_j I_n) \mathbf{v}_{k-m}$;
        $s_{k-m}(z_j) = h_{k-m+1,k-m} / \sqrt{h_{k-m+1,k-m}^2 + |\tau_{k-m}(z_j)|^2}$;
        $c_{k-m}(z_j) = \tau_{k-m}(z_j) / \sqrt{h_{k-m+1,k-m}^2 + |\tau_{k-m}(z_j)|^2}$;
        $\mathbf{w}_{k-m}(z_j) = s_{k-m}(z_j) \mathbf{w}_{k-m-1}(z_j) + (-1)^{k-m} \overline{c_{k-m}(z_j)} \mathbf{v}_{k-m+1}$;
    end
```

Note that it is numerically more stable if we do not divide by the square root $\sqrt{h_{k-m+1,k-m}^2 + |\tau_{k-m}(z_j)|^2}$ in the computation of $s_{k-m}(z_j)$ and $c_{k-m}(z_j)$, but instead normalize $\mathbf{w}_{k-m}(z_j)$ after each iteration (this however, leads to m_2 additional scalar products). If we assume the matrix under consideration is sparse; allowing a computational complexity of $\mathcal{O}(n)$ to compute a matrix vector product; the total complexity to compute the first k orthonormal Arnoldi vectors can be estimated as $\mathcal{O}(kn)$.

Remark 3.8. *If the rational function in* (1.1) *has only one pole, i.e.,* $m_2 = 1$ *in* (3.6) *then the order in which the coefficients are determined can be reversed. More precisely, we can first compute* $a_{1,k}$ *as* $\mathbf{w}_{k-m-1}(z_1)^* \mathbf{v}_k'$ *and then orthonormalize the resulting difference* $\mathbf{v}_k' - a_{1,k} \mathbf{w}_{k-m-1}(z_1)$ *against* $\mathbf{v}_{k-m+1}, \ldots, \mathbf{v}_k$ *to obtain* $\mathbf{v}_{k+1}$. *This might be of influence on the numerical performance.*

```
Data: $A \in \mathbb{C}^{n\times n}$ with $A^* = \sum_{j=1}^{m_2} d_j(A - z_jI_n)^{-1} + \pi(A)$, $\pi$ of degree $m_1 - m_2$, $F = [\mathbf{f}_1, \dots, \mathbf{f}_{m_3}]$,
      $G = [\mathbf{g}_1, \dots, \mathbf{g}_{m_3}] \in \mathbb{C}^{n\times m_3}$, $\mathbf{b} \in \mathbb{C}^n$, $i$
Result: $V_{i+1} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{i+1}] \in \mathbb{C}^{n\times(i+1)}$
$m := \max(0, m_1 - m_2 + 1)$;
$\mathbf{v}_1 = \mathbf{b}/||\mathbf{b}||$;
for $j = 1, \dots, m_2$ do
    $\mathbf{w}_0(z_j) = \mathbf{v}_1$;
end
for $k = 1, \dots, m$ do
    $\mathbf{v}' = A\mathbf{v}_k$;
    for $j = 1, \dots, i$ do
        $h_{j,k} = \mathbf{v}_j^*\mathbf{v}'$; $\mathbf{v}' = \mathbf{v}' - h_{j,k}\mathbf{v}_j$;
    end
    $h_{k+1,k} = ||\mathbf{v}'||$; $\mathbf{v}_{k+1} = \mathbf{v}'/h_{k+1,k}$;
end
$\widetilde{G} := 0 \in \mathbb{C}^{n\times m_3}$;
for $k = m+1, \dots, i$ do
    $\hat{F}_{k,:} := \mathbf{v}_k^*F$; $\hat{G}_{k-m,:} := \mathbf{v}_{k-m}^*G$;
    $\widetilde{G} := \widetilde{G} + \mathbf{v}_{k-m}\hat{G}_{k-m,:}$;
    $\mathbf{v}' := A\mathbf{v}_k - \widetilde{G}\hat{F}_{k,:}^*$;
    for $j = k, k-1, \dots, k-m+1$ do
        $h_{j,k} := \mathbf{v}_j^*\mathbf{v}'$; $\mathbf{v}' := \mathbf{v}' - h_{j,k}\mathbf{v}_j$;
    end
    $Q := [\mathbf{q}_1, \dots, \mathbf{q}_{m_2}], R := (r_{i,j}) \in \mathbb{C}^{m_2\times m_2}$, such that $QR = [\mathbf{w}_{k-m-1}(z_1), \dots, \mathbf{w}_{k-m-1}(z_{m_2})]$;
    for $j = m_2, m_2 - 1, \dots, 1$ do
        $a_{j,k} := \left(\mathbf{q}_j^*\mathbf{v}' - \sum_{\ell=j+1}^{m_2} r_{j,\ell}a_{j,k}\right)/r_{j,j}$;
        $\mathbf{v}' = \mathbf{v}' - a_{j,k}\mathbf{w}_{k-m-1}(z_j)$;
    end
    $h_{k+1,k} = ||\mathbf{v}'||$; $\mathbf{v}_{k+1} := \mathbf{v}'/h_{k+1,k}$;
    for $j = 1, \dots, m_2$ do
        $\tau_{k-m}(z_j) = (-1)^{k-m-1}\mathbf{w}_{k-m-1}(z_j)^*(A - z_jI_n)\mathbf{v}_{k-m}$;
        $s_{k-m}(z_j) = h_{k-m+1,k-m}/\sqrt{h_{k-m+1,k-m}^2 + |\tau_{k-m}(z_j)|^2}$;
        $c_{k-m}(z_j) = \tau_{k-m}(z_j)/\sqrt{h_{k-m+1,k-m}^2 + |\tau_{k-m}(z_j)|^2}$;
        $\mathbf{w}_{k-m}(z_j) = s_{k-m}(z_j)\mathbf{w}_{k-m-1}(z_j) + (-1)^{k-m}\overline{c_{k-m}(z_j)}\mathbf{v}_{k-m+1}$ ;
    end
end
```

Algorithm 1: The fast Arnoldi method

4 Connection with the Barth-Manteuffel multiple recurrence relation

The aim of this section is to show how our work is related to that of Barth & Manteuffel in their article on 'Multiple recursion conjugate gradient algorithms' [2]. They introduce an economical conjugate gradient algorithm for the class of BML-matrices, by making use of short recurrence relations. We will give a short summary of their findings and discuss both the differences and similarities with our approach.

To prevent the possibility of a breakdown in their so-called 'single recurrence relation' Barth & Manteuffel rewrote it as a set of recurrence relations, that are stated in (4.1)-(4.3).

$$\begin{aligned}\underline{\mathbf{p}}_{j+1} &= A\underline{\mathbf{p}}_j - \sum_{i=j-(\ell-m)}^{j} t_{i,j}\underline{\mathbf{p}}_i - [\underline{\mathbf{q}}_{j_0},\dots,\underline{\mathbf{q}}_{j_{m-1}}]\underline{\eta}_j \\ &\quad - [\underline{\hat{\mathbf{q}}}_{j_0},\dots,\underline{\hat{\mathbf{q}}}_{j_{\kappa-1}}]\underline{\mu}_j, && (4.1)\\ \underline{\mathbf{q}}_{j+1_i} &= \frac{(\overline{\underline{\rho}}_{j+1})_i}{\underline{\mathbf{p}}_{j+1}^*\underline{\mathbf{p}}_{j+1}}\underline{\mathbf{p}}_{j+1} + \underline{\mathbf{q}}_{j_i} \qquad \text{for} \qquad i = 0,\dots,m-1, && (4.2)\\ \underline{\hat{\mathbf{q}}}_{j+1_i} &= \frac{(\overline{\underline{\tau}}_{j+1})_i}{\underline{\mathbf{p}}_{j+1}^*\underline{\mathbf{p}}_{j+1}}\underline{\mathbf{p}}_{j+1} + \underline{\hat{\mathbf{q}}}_{j_i} \qquad \text{for} \qquad i = 0,\dots,\kappa-1. && (4.3)\end{aligned}$$

Unfortunately, they use different letters, shift indices, and construct an orthogonal, but not orthonormal basis of the Krylov space. The new normalization comes from the fact that they consider a Hessenberg matrix which has ones on its first subdiagonal. Their basis vectors $\underline{\mathbf{p}}_0, \underline{\mathbf{p}}_1, \dots, \underline{\mathbf{p}}_{N-1} \in \mathbb{C}^n$ satisfy $\underline{\mathbf{p}}_j / \|\underline{\mathbf{p}}_j\| = \mathbf{v}_{j+1}$. Moreover, they use the integers $(\ell, m, \kappa, \theta)$ instead of $(m_1, m_2, m_3, m_3 - 1 + m)$. Also, as seen in (4.1)-(4.3), two other families of vectors with double indices are used. For simplicity and consistency we will abbreviate them as

$$W_k := [\underline{\mathbf{q}}_{k_0}, ..., \underline{\mathbf{q}}_{k_{m_2-1}}] \in \mathbb{C}^{n\times m_2} \qquad \text{and} \qquad \widehat{W}_k := [\underline{\hat{\mathbf{q}}}_{k_0}, ..., \underline{\hat{\mathbf{q}}}_{k_{m_3-1}}] \in \mathbb{C}^{n\times m_3}.$$

As the reader will see below, to compare our approach with [2], we will not explicitly make use of (4.1)-(4.3), but instead make use of a mathematical equivalent of (4.1)-(4.3) which is adapted to our notation and scalings. The original pseudocode used by Barth & Manteuffel is stated in Algorithm 2.

In [2, Eqn. (4.16)] the authors provide an explicit formula for the entries of the upper Hessenberg matrix H:

$$H_{j,k} = \mathbf{v}_j^* A \mathbf{v}_k = \underline{\rho}_j^* \underline{\eta}_k + \underline{\tau}_j^* \underline{\mu}_k, \ \underline{\rho}_j, \underline{\eta}_k \in \mathbb{C}^{m_2}, \ \underline{\mu}_j, \underline{\tau}_k \in \mathbb{C}^{m_3}, \tag{4.4}$$

for all $j = 1, 2, ..., k - m$, in which the reader recognizes generators $\underline{\rho}_i, \underline{\eta}_i, \underline{\mu}_i$ and $\underline{\tau}_i$ for the low-rank part of H. Note that, in contrast to [2], we start numbering with $i = 1$ instead of $i = 0$. To be able to use the recurrence relations (4.1)-(4.3) in practice, the generators $\underline{\rho}_i, \underline{\eta}_i, \underline{\mu}_i$ and $\underline{\tau}_i$ need to be known in advance. Therefore, Algorithm 2 is based on a rewritten form of the recurrence relations (4.1)-(4.3) which enables to compute the orthogonal basis vectors $\mathbf{p}_i$ and the generators $\underline{\rho}_i, \underline{\eta}_i, \underline{\mu}_i$ and $\underline{\tau}_i$ simultaneously.

We define W_k and $\widehat{W}_k$ [2, Eqn. (4.22) and Eqn. (4.23)] as

$$W_k = \sum_{j=1}^{k+1} \mathbf{v}_j \underline{\rho}_j^* \in \mathbb{C}^{n\times m_2}, \quad \widehat{W}_k = \sum_{j=1}^{k+1} \mathbf{v}_j \underline{\tau}_j^* \in \mathbb{C}^{n\times m_3}, \tag{4.5}$$

for $k = 0, \dots, n-1$, which is the equivalent of (4.2)-(4.3). With a similar reasoning as in [2, Eqn. (5.3)] it can be proven that

$$A\mathbf{v}_k = \sum_{j=1}^{k+1} \mathbf{v}_j H_{j,k} = \sum_{j=k-m+1}^{k+1} \mathbf{v}_j H_{j,k} + W_{k-m-1}\underline{\eta}_k + \widehat{W}_{k-m-1}\underline{\mu}_k, \tag{4.6}$$

which is mathematically equivalent to (4.1).

Suppose now that the generators as defined in (4.4) are known. Then one can use (4.6) to compute $\mathbf{v}_{k+1}$ out of $\mathbf{v}_{k-m+1}, ..., \mathbf{v}_k, W_{k-m-1}$ and $\widehat{W}_{k-m-1}$, then use (4.5) to compute $W_{k-m}, \widehat{W}_{k-m}$ out of $\mathbf{v}_{k-m+1}, W_{k-m-1}$ and $\widehat{W}_{k-m-1}$, and so on. Hence, it remains to find the generators. Two of them can be computed with an explicit formula [2, Eqn. (4.15) and (4.13)], namely

$$\underline{\mu}_k = F^* \mathbf{v}_k, \quad \underline{\eta}_k = (H_{j,k})_{j=1,\dots,m_2}. \tag{4.7}$$

The vector $\underline{\rho}_j$ is obtained [2, Eqn. (4.12)] as the 'remainder' in the polynomial division of q_{j-1} (2.6) by the denominator q (1.1):

$$q_{j-1}(z) = \alpha_j(z) q(z) + (q_0(z), ..., q_{m_2-1}(z))\underline{\rho}_j, \tag{4.8}$$

where we observe that $\rho_1, ..., \rho_{m_2}$ form the canonical basis of $\mathbb{C}^{m_2}$, and thus $V_{m_2}^* W_k = I$. We may rewrite the remainder in terms of the Lagrange polynomials ℓ_h of the roots $z_1, .., z_{m_2}$ of q, leading to

$$(q_0(z), ..., q_{m_2-1}(z))\underline{\rho}_j = \sum_{h=1}^{m_2} q_{j-1}(z_h)\ell_h(z) \quad \text{or} \quad \underline{\rho}_j = \sum_{h=1}^{m_2} q_{j-1}(z_h) V_{m_2}^* \ell_h(A)\mathbf{v}_1.$$

Substituting the expression for $\underline{\rho}_j$ into (4.5) allows to conclude that

$$W_k = M_k \begin{bmatrix} \sigma_k(z_1)\mathbf{v}_1^*\ell_1(A)^* V_{m_2} \\ \vdots \\ \sigma_k(z_{m_2})\mathbf{v}_1^*\ell_{m_2}(A)^* V_{m_2} \end{bmatrix}. \tag{4.9}$$

Recalling that $V_{m_2}^* W_k = I$ gives

$$W_k = M_k \Big(V_{m_2}^* M_k\Big)^{-1}, \tag{4.10}$$

which makes a partial link between (4.6) and Proposition 3.4. In particular, if the matrix A is unitary and no low-rank perturbation is involved, W_k is a multiple of $\mathbf{w}_k(0)$ and the Barth-Manteuffel multiple recurrence relation turns out to be equivalent to the Szegő recurrence relations. The quantities $\underline{\rho}_{k+1}$ are computed recursively [2, Eqn. (5.11)] by computing all entries of $\underline{H}_k$, and by taking remainders after division by q in the relation $zq_{k-1}(z) = \sum_{j=1}^{k+1} H_{j,k} q_{j-1}(z)$, the polynomial translation of the Arnoldi relation $AV_k = V_{k+1}\underline{H}_k$. We refer to lines $6, 8, 12, 25, 27$ and 36 of Algorithm 2.

It remains to show how to compute $\underline{\tau}_{k-m+1}$ (after having computed $\mathbf{v}_{k+1}, \rho_{k-m+1}, W_{k-m}$) and relate the term $\widehat{W}_{k-m-1}\underline{\mu}_k$ in (4.6) to the term $V_{k-m}G_{k-m}F^*\mathbf{v}_k = V_{k-m}V_{k-m}^* G\underline{\mu}_k$ of our short recurrence of Proposition 3.4. In fact, at this place the authors of [2] require an additional delay in the recurrence (4.6) by replacing m by $m' := m+m_3 \geq m$, which is possible according to (4.4). According to [2, Eqn. (5.2)] and (4.4), the computation of $\underline{\tau}_{k-m'+1}$ is done by solving the system

$$\underline{\tau}_{k-m'+1}^* [\underline{\mu}_{k-\ell}]_{\ell=0,\dots,m_3-1} = [H_{k-m'+1,k-\ell} - \underline{\rho}_{k-m'+1}^* \underline{\eta}_{k-\ell}]_{\ell=0,\dots,m_3-1}. \tag{4.11}$$

We refer to line 16 of Algorithm 2. However, there is a possible problem with this system which is not discussed in [2]. As noticed after [2, Eqn. (5.2)], it is consistent, but one may not insure that the matrix is invertible, i.e., we might have several solutions, each of them being a generator suitable for H_k, but not necessarily for H_N. This is a general problem with computing generators for H_N in a recursive manner: there is no guarantee that $m_3 = \text{rank}(F^*)$ is equal to $\text{rank}(F^*V_N)$, and thus whether the minimal number of generators for H_N is equal to $m_2 + m_3$. In addition, at stage k of a recursive computation, it might happen that the minimal number of generators for H_k is strictly lower, i.e., $\text{rank}(F^*V_k) < \text{rank}(F^*V_N)$, i.e., we should have a m_3 depending on k. Finally, the matrix of coefficients is just obtained by picking the last m_3 columns of F^*V_k which might also lower the rank. However, going through the proof of (4.4) we can derive an explicit formula for $\underline{\tau}_j$. From [2, Eqn. (4.14)] it can be deduced that

$$\underline{\tau}_j = G^* q(A)\alpha_j(A)\mathbf{v}_1. \tag{4.12}$$

Combining (4.12) and (4.8), we obtain

$$\begin{aligned} \underline{\tau}_j &= G^*\Big(\mathbf{v}_j - (\mathbf{v}_1, \dots, \mathbf{v}_{m_2-1})\underline{\rho}_j\Big) \\ &= G^*\mathbf{v}_j - G^*V_{m_2}\underline{\rho}_j. \end{aligned} \tag{4.13}$$

Note that (4.13) could be used to compute $\underline{\tau}_{k-m+1}$ without introducing the additional delay in (4.6). Inserting (4.13) into (4.5) gives the explicit formula

$$\begin{aligned} \widehat{W}_{k-m-1} &= \sum_{j=1}^{k-m} \mathbf{v}_j \Big(\mathbf{v}_j^* G - \underline{\rho}_j^* V_{m_2}^* G\Big) \\ &= V_{k-m}V_{k-m}^* G - W_{k-m-1}V_{m_2}^* G. \end{aligned}$$

In the following remark we intend to compare our approach with that of Barth & Manteuffel [2].

Remark 4.1. *(a) Both approaches heavily use the fact that a certain upper right part of the Hessenberg matrix is of rank at most $m_2 + m_3$. In other words, one is able to express $A\mathbf{v}_k$ as a linear combination of $m_2 + m_3$ correction vectors plus a linear combination of the last m or $m' = m + m_3$ columns of V_{k+1}, a kind of corrected "short" recurrence. Notice however that our recurrence is "shorter" if $m_3 > 0$.*

(b) In our approach, m_3 correction vectors are explicitly given (the term $\mathbf{v}'_k - A\mathbf{v}_k$) and can be updated explicitly. They are not necessarily linearly independent. The other m_2 correction vectors are identified as GMRES residuals for shifted systems, allowing for easy updating.

In contrast, Barth & Manteuffel compute explicitly the four sequences of generators of the low-rank structure of H_N, given in (4.4)*. Notice, however, that at the kth step of the algorithm one can only deduce generators for H_k and not for H_N. As mentioned above, in order not to be obliged to correct generators found earlier, there should be an additional assumption on rank$(F^* V_k)$ not mentioned by the authors. However, there is a variant of the Barth & Manteuffel approach: instead of using* (4.11) *requiring a delay in the recurrence relation* (4.6)*, one can use* (4.13)*, which is not mentioned in [2], to compute $\underline{\tau}_{k-m+1}$ just after having computed $\underline{\rho}_{k-m+1}$.*

(c) In our approach, for finding the coefficients of the GMRES correction vectors, we suggest to solve a least square problem, the matrix of coefficients M_k having as columns these m_2 (normalized) GMRES correction vectors. Notice that M_k has full column rank (since $V_{m_2}^* M_k$ has), but might be ill-conditioned. Thus standard techniques (SVD dropping small singular values, or QR decomposition with column pivoting and threshold) can be applied, where the residual error in solving this least-square problem leads to a loss of orthogonality for v_{k+1} of the same order.

In contrast, Barth & Manteuffel suggest one of the missing generators by solving system (4.11)*. The computation of the other generator $\underline{\rho}_{j+1}$ is quite involved and requires the knowledge of the whole jth column of the Hessenberg matrix $\underline{H}_k$ (which is not necessarily computed using our approach).*

(d) If GMRES is converging fast, we believe that the normalization (4.10) *is not appropriate since*

$$\|V_{m_2}^* M_k\| \ll \|M_k\| \approx 1.$$

Also the $\underline{\eta}_k$ are very small due to the above-mentioned decay property of the entries of our Hessenberg matrix.

5 Some special matrices

In this section we give special attention to some classes of matrices where we can slightly reduce the computational complexity of Algorithm 1. The first class consists of matrices A which satisfy an equation of the form

$$A^* - \alpha A - \beta I = FG^*, \tag{5.1}$$

for some $\alpha, \beta \in \mathbb{C}$. This includes the class of normal matrices of which all but m_3 eigenvalues are collinear. We will address this kind of matrices as *nearly Hermitian matrices.* If $\alpha = 1, \beta = 0$ and A is real, this corresponds to the class of *nearly symmetric matrices* as discussed in [4]. However, one can easily check that all results derived in [4] are also valid for a matrix of the form (5.1).

The second class consists of matrices A which satisfy an equation of the form

$$A^* - \alpha I - \beta(A - \delta I)^{-1} = FG^*, \tag{5.2}$$

for some $\alpha, \beta, \delta \in \mathbb{C}$. This includes the class of normal matrices of which all but m_3 eigenvalues are concyclic. If $\delta = 0$, we speak of *nearly unitary matrices*, if $\delta \neq 0$ we speak of *nearly shifted unitary matrices.*

The class of matrices satisfying (3.5) or (5.2) include all examples of BML-matrices known to us which are of practical interest. By this we mean, matrices that are suitably large with respect to the quantities m_1, m_2 and m_3. More information on matrices satisfying equation (1.1) can be found in [13].

5.1 Nearly Hermitian matrices

Assume the matrix A is nearly Hermitian. Then (3.6) reduces to

$$\mathbf{v}_k' = h_{k+1,k}\mathbf{v}_{k+1} + h_{k,k}\mathbf{v}_k + h_{k-1,k}\mathbf{v}_{k-1}. \tag{5.3}$$

Define the vectors $\mathbf{p}_k^* \in \mathbb{C}^{m_3}$ recursively as

$$\mathbf{p}_k^* := -s_{k-1}(0)\mathbf{p}_{k-1}^* + c_{k-1}(0)\mathbf{e}_k^* G_k, \tag{5.4}$$

for $k \geq 2$ and $\mathbf{p}_1^* = G_1$. By recurrence on k it follows that $\mathbf{p}_k^* = \mathbf{e}_k^* Q_k(0)^* G_k$. Therefore, in combination with (2.25),

$$\mathbf{p}_k^* = (-1)^{k-1}\mathbf{w}_{k-1}(0)^* V_k G_k. \tag{5.5}$$

Then (2.23) yields

$$\begin{aligned}
\tau_k(0) &= (-1)^{k-1}\mathbf{w}_{k-1}(0)^* A\mathbf{v}_k \\
&= (-1)^{k-1}\mathbf{w}_{k-1}(0)^*(\mathbf{v}_k' + V_k G_k F_k^* \mathbf{e}_k) \\
&= (-1)^{k-1}\mathbf{w}_{k-1}(0)^*\mathbf{v}_k' + (-1)^{k-1}\mathbf{w}_{k-1}(0)^* V_k G_k F_k^* \mathbf{e}_k \\
&= h_{k,k}c_{k-1}(0) - h_{k-1,k}c_{k-2}(0)s_{k-1}(0) + \mathbf{p}_k^* F_k^* \mathbf{e}_k,
\end{aligned} \tag{5.6}$$

the latter equality because of (5.3) and (5.5). Expression (5.6) can now be used to compute $\tau_k(0)$ instead of (2.23), reducing the computational complexity[2].

5.2 Nearly unitary matrices

Assume the matrix A is nearly unitary. Then (3.6) reduces to

$$\mathbf{v}_k' = a_{1,k}\mathbf{w}_{k-1}(0) + h_{k+1,k}\mathbf{v}_{k+1}, \tag{5.7}$$

where $a_{1,k} = \mathbf{w}_{k-1}(0)^*\mathbf{v}_k'$ and $h_{k+1,k}$ such that $\mathbf{v}_{k+1}$ is of unit length. Again we make use of the vector $\mathbf{p}_k^*$ as defined in (5.4). Then because of (5.5), (2.23) yields

$$\begin{aligned}
\tau_k(0) &= (-1)^{k-1}\mathbf{w}_{k-1}(0)^* A\mathbf{v}_k \\
&= (-1)^{k-1}\mathbf{w}_{k-1}(0)^*(\mathbf{v}_k' + V_k G_k F_k^* \mathbf{e}_k) \\
&= (-1)^{k-1}a_{1,k} + \mathbf{p}_k^* F_k^* \mathbf{e}_k.
\end{aligned} \tag{5.8}$$

Expression (5.8) can now be used to compute $\tau_k(0)$ instead of (2.23).

5.3 Nearly shifted unitary matrices

Assume the matrix A is nearly shifted unitary. Then (3.6) reduces to

$$\mathbf{v}_k' = a_{1,k}\mathbf{w}_{k-2}(\delta) + h_{k,k}\mathbf{v}_k + h_{k+1,k}\mathbf{v}_{k+1}, \tag{5.9}$$

where $a_{1,k} = \mathbf{w}_{k-2}(\delta)^*\mathbf{v}_k'$ and $h_{k,k}$, $h_{k+1,k}$ are entries of the corresponding Hessenberg matrix. From (2.22) we deduce that

$$\mathbf{v}_{k-1}^*\mathbf{w}_{k-2}(\delta) = (-1)^{k-2}\overline{c_{k-2}(\delta)}. \tag{5.10}$$

Hence, due to (2.22) and (5.9),

$$\begin{aligned}
\mathbf{w}_{k-2}(\delta)^*\mathbf{v}_{k-1}' &= s_{k-2}(\delta)\mathbf{w}_{k-3}(\delta)^*\mathbf{v}_{k-1}' + (-1)^{k-2}c_{k-2}(\delta)\mathbf{v}_{k-1}^*\mathbf{v}_k' \\
&= s_{k-2}(\delta)a_{1,k-1} + (-1)^{k-2}c_{k-2}(\delta)a_{1,k}\mathbf{v}_{k-1}^*\mathbf{w}_{k-2}(\delta) \\
&= s_{k-2}(\delta)a_{1,k-1} + a_{1,k}|c_{k-2}(\delta)|^2.
\end{aligned} \tag{5.11}$$

Finally, we know that

$$(-1)^{k-1}\mathbf{w}_k(\delta)^* V_k G_k = s_k(\delta)\mathbf{p}_k^*(\delta), \tag{5.12}$$

with $\mathbf{p}_k^*(\delta) \in \mathbb{C}^{m_3}$ recursively defined as

$$\mathbf{p}_k^* := -s_{k-1}(\delta)\mathbf{p}_{k-1}^* + c_{k-1}(\delta)\mathbf{e}_k^* G_k, \tag{5.13}$$

[2] Expression (5.6) was also proved alternatively By Beckermann and Reichel [4, Proposition 4.2].

for $k \geq 2$ and $\mathbf{p}_1^* = G_1$. Making use of (5.10), (5.11) and (5.12), (2.23) yields

$$\begin{aligned}
\tau_{k-1}(\delta) &= (-1)^{k-2}\mathbf{w}_{k-2}(\delta)^*(A - \delta I_n)\mathbf{v}_{k-1} \\
&= (-1)^{k-2}\mathbf{w}_{k-2}(\delta)^* \left(\mathbf{v}'_{k-1} + V_{k-2}G_{k-2}F_{k-1}^*\mathbf{e}_{k-1} - \delta\mathbf{v}_{k-1}\right) \\
&= (-1)^{k-2}s_{k-2}(\delta)a_{1,k-1} + (-1)^{k-2}a_{1,k}|c_{k-2}(\delta)|^2 \\
&\quad -s_{k-2}(\delta)\mathbf{p}_{k-2}(\delta)^*F_{k-1}^*\mathbf{e}_{k-1} - \delta c_{k-2}(\delta).
\end{aligned}$$

As before, the above expression can now be used to compute $\tau_{k-1}(\delta)$ instead of (2.23).

6 Numerical examples

In this section we will compare our fast Arnoldi algorithm with the one of Barth-Manteuel and classical Arnoldi. We focus especially on the orthogonality of the obtained Arnoldi vectors. The orthogonality in the forthcoming figures is measured by a method described originally by Paige [6, 15]. Given $V_k^*V_k - I = U_k + U_k^*$ with U_k strictly upper triangular, we define $S_k = (I + U_k)^{-1}U_k$. The norm of S_k is used as an orthogonality measure for the columns of V_k, i.e., $\|S_k\| \in [0, 1]$ where $\|S_k\| = 0$ when they are orthonormal and $\|S_k\| = 1$ when they are linearly dependent [15]. To make the fairest possible comparison with the Barth-Manteuffel algorithm, we implemented their pseudocode as stated in [2] and recalled in Algorithm 2. However, their pseudocode returns an orthogonal basis, while recurrence relation (3.6) returns an orthonormal basis. Therefore, we have normalized these vectors first.

We will start in §6.1 and §6.2 by discussing the special case of nearly unitary A (with $\delta = 0$) and nearly shifted unitary A (with shift $\delta \neq 0$), where we replaced in our BML-Arnoldi algorithm formula (2.23) for the computation of $\tau_k(\delta)$ by the less expensive formulas described in §5.2, and §5.3, respectively. Subsequently we report in §6.3 about an example of a nearly Hermitian matrix discussed already in [6].

Quite often there is some correlation between loss of orthogonality between Arnoldi vectors and convergence of the GMRES residual $r_k(\delta)$ of the shifted system $(A-\delta I)x = b$ with starting vector $x_0 = 0$. This phenomenon is probably related to the decay properties mentioned in Remark 2.7. We therefore draw in each of the figures below the relative GMRES residual

$$\frac{\|r_k(\delta)\|}{\|r_0(\delta)\|} = \frac{1}{\sigma_k(\delta)} = \prod_{j=1}^{k} s_j(\delta), \tag{6.1}$$

the last identity following from (2.8). Notice that the quantities $s_j(\delta)$ are already computed in the BML-Arnoldi algorithm in the case $m_3 = 1$ of §6.1 and §6.2, whereas in §6.3 we have to add the computation of $s_j(\delta)$, here for $\delta = 0$, following the formulas given in §5.1. One may understand (6.1) as the recursive computation of the GMRES residuals following some progressive residual scheme, where the underlying least squares problem is solved by successive Givens rotations. We will refer to this residual in the forthcoming figures as the *progressive residual*. However, due to loss of orthogonality, it might be that these progressive residuals are badly computed. This is why each time we display also the "exact" relative GMRES residual, obtained by computing the kth iterate of GMRES for the shifted system $(A-\delta I)x = b$ with starting vector $x_0 = 0$ via the black box routine of Matlab (which does not use our Arnoldi vectors but recomputes them via full Arnoldi, and solves the least squares problem via Householder transforms). It turns out that, in all our numerical experiments, that when both Arnoldi and fast Arnoldi behave well, that the progressive residual and the GMRES residual exhibit the same convergence history.

All computations were carried out in Matlab R2015a. As a starting vector for the Krylov subspace we always consider a vector b that has normally distributed random entries with mean zero and variance one.

6.1 Perturbed diagonal and unitary matrices

We consider 200×200 diagonal matrices for which all but m_3 eigenvalues lie on a circle. Clearly, such matrices satisfy equation (1.1) with $m_1 = m_2 = 1$. We considered various cases; for each case we show the eigenvalues of the matrix and a comparison of the orthogonality of the computed Arnoldi vectors for classical Arnoldi, Barth-Manteuffel, and the fast Arnoldi method. The legend is plotted in Figure 1 and is identical for all similar graphs in this section.

1. In the first experiment, see Figure 1, we consider eigenvalues on three quarter of the unit circle. Clearly the full Arnoldi and fast Arnoldi perform best and in all the tests we ran the orthogonality of the computed vectors was comparable. Other experiments revealed that Barth-Manteuffel performed just slightly worse when considering eigenvalues distributed over the entire unit circle, the performance of Barth-Manteuffel started to degrade when segments were excluded from the unit circle. The progressive residual seems to align almost perfectly with the GMRES residual. We have also tested various radii and similar conclusions hold when the radius of the circle is changed.
2. In the second experiment, see Figure 2, we have shifted the unit circle in the complex plane. Barth-Manteuffel seems to have problems with this case, Arnoldi, and the fast Arnoldi method on the other hand exhibit good accuracy. The progressive residual and the GMRES residual align again almost perfectly.
3. In a third experiment, see Figure 3, all eigenvalues except for two are located on the unit circle. In this case Barth-Manteuffel outperforms our code slightly. We note that the location of the eigenvalues outside the circle does not have a significant impact on the overall picture of the accuracy. Tests revealed also that if one would shift the midpoint or exclude eigenvalues out of parts of the circle the fast Arnoldi method would outperform Barth-Manteuffel.

Overall we can conclude that the Arnoldi method is the most accurate one and the fast Arnoldi method is also typically quite close. The Barth-Manteuffel algorithm, however, exhibits quick loss of orthogonality when the circle is shifted or the eigenvalues do not span the entire circle. Moreover, the progressive residual and the GMRES residual align almost perfectly.

We also considered a real non-normal matrix A which is of the form

$$A = U + \mathbf{u}\mathbf{v}^*, \tag{6.2}$$

for some vectors $\mathbf{u}$ and $\mathbf{v}$ and a unitary matrix U. Using the Sherman-Morrison formula [9], equation (6.2) yields

$$A^* = A^{-1} + \mathbf{v}\mathbf{u}^* + \frac{U^*\mathbf{u}\mathbf{v}^*U^*}{1+\mathbf{v}^*U^*\mathbf{u}}.$$

Hence,

$$A^* = A^{-1} + FG^*, \text{ with } F := \begin{pmatrix} \mathbf{v} & \frac{1}{1+\mathbf{v}^*U^*\mathbf{u}}U^*\mathbf{u} \end{pmatrix}, \; G := \begin{pmatrix} \mathbf{u} & U\mathbf{v} \end{pmatrix}.$$

The orthogonality of the computed Arnoldi vectors was examined for a 100×100 matrix A of the form (6.2) where the unitary matrix U and the vectors $\mathbf{u}$ and $\mathbf{v}$ are randomly generated. In this case the orthogonality behaved similar to Figure 1, implying that the fast Arnoldi behaves similar to Arnoldi and Barth-Manteuffel deteriorates.

6.2 Unitary matrix from Quantum Chromodynamics

We consider a shifted unitary matrix which finds its origin in Quantum Chromodynamics (QCD) [1]. QCD is the theory which describes the fundamental interaction between quarks, which are the building blocks of protons and neutrons. This theory makes use of the Neuberger overlap operator $A = \rho I + \gamma\,\mathrm{sign}(Q)$, where ρ and γ are scalars and Q is the Hermitian Wilson fermian matrix. As a result the Neuberger overlap operator is a shifted unitary matrix. To construct the matrix Q a parameter κ and a hopping matrix are needed. We have selected these parameters equal to the ones from [1], i.e. $\kappa = 0.2809$ and as hopping matrix `conf5.0-00.14x4-2600.mtx` from the Matrix Market[3].

To compute $\mathrm{sign}(Q)$, we invoke the software package designed by Arnold, et al., [1], which makes use of the Zolotarev algorithm [17]. We choose parameters $\rho = 2$ and $\gamma = 1$ for the Neuberger overlap operator. On the left of Figure 4 we have drawn the eigenvalues of the Neuberger overlap operator A and observe that the density of the eigenvalues is much higher on the right than on the left.

[3] A repository of test data for use in comparative studies of algorithms for numerical linear algebra, featuring nearly 500 sparse matrices from a variety of applications, as well as matrix generation tools and services.

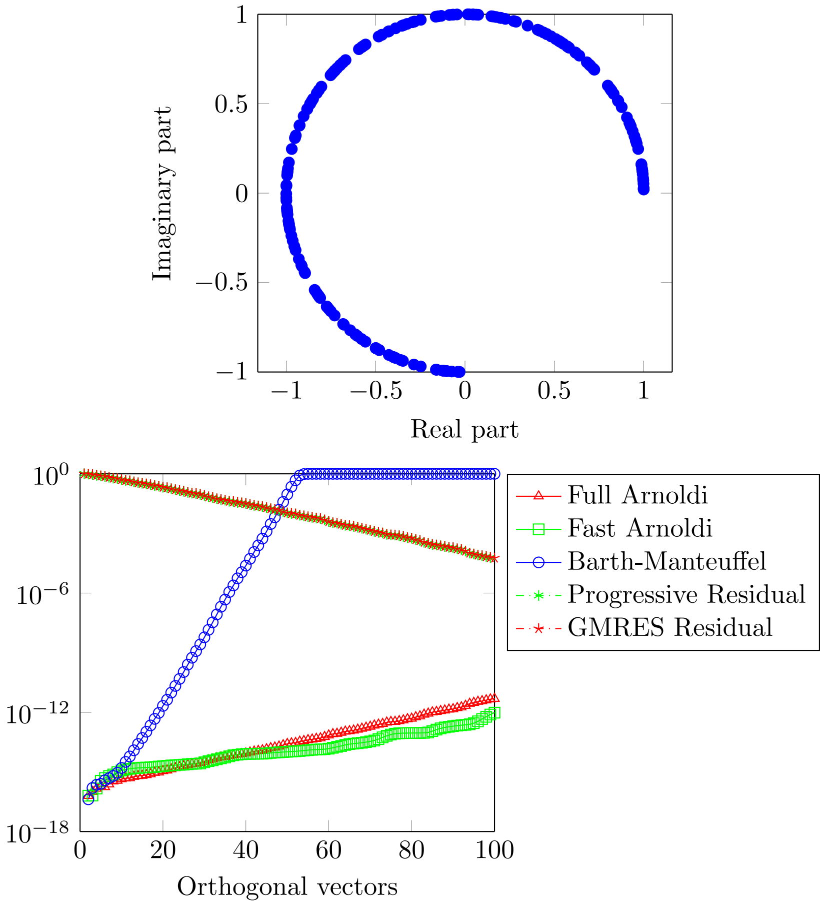

Figure 1: The orthogonality, measured by $\|S_k\|$ as in [15], of the computed Arnoldi vectors for classical Arnoldi, Barth-Manteuffel, and the fast Arnoldi, for a matrix having its eigenvalues randomly distributed over three quarter of the unit circle. The progressive residual as well as the residual of GMRES are also depicted, we observe an almost identical behavior for both residuals.

The orthogonality of the computed Arnoldi vectors is depicted in Figure 4. Approximately the first ten iterations of the recurrence relation (3.6) fast Arnoldi and the Barth-Manteuffel algorithm show comparable accuracy. After that, the orthogonality of the vectors computed with the Barth-Manteuffel algorithm deteriorates fast at almost the same rate as classical Arnoldi. Even though the progressive residuals and the GMRES residual align, classical Arnoldi seems to suffer heavily from loss of accuracy. The fast Arnoldi method clearly outperforms the other approaches.

6.3 Departure from orthogonality

Beckermann & Reichel [4] proposed a Krylov subspace method for solving a linear system in which the coefficient matrix is nearly Hermitian. Their method, based on a short recurrence for generating an orthonormal Krylov basis, is better known as *Progressive GMRES*, shortly named PGMRES. For nearly Hermitian matrices, this short recurrence coincides with the recurrence relation (3.6) described above.

However, Embree, et al., [6] showed how in certain cases the PGMRES method exhibits an instability which finds its origin in the loss of orthogonality of the computed Arnoldi vectors. A specific class of examples is described and the corresponding departure from orthogonality is shown when using the recurrence relation (3.6). In the forthcoming experiments we have used the algorithm for nearly Hermitian matrices as presented in Section 5.1.

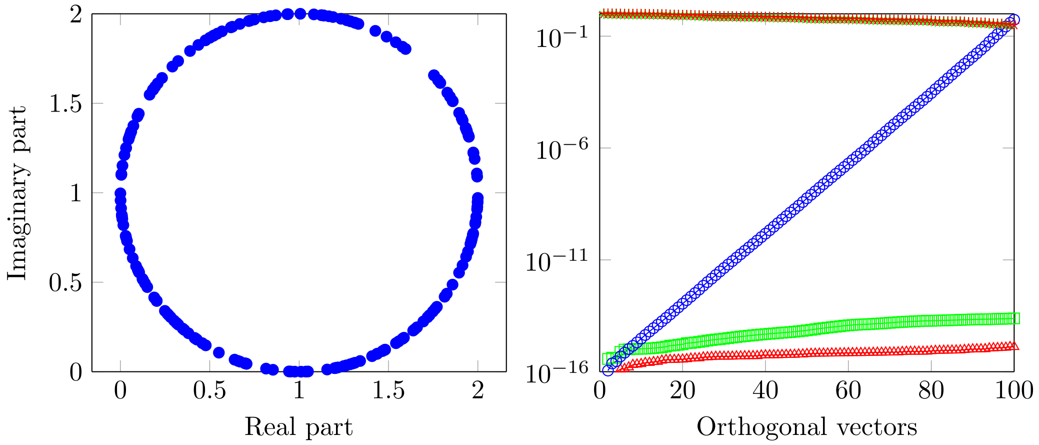

Figure 2: The orthogonality, measured as in [15], of the computed Arnoldi vectors for classical Arnoldi, Barth-Manteuffel, and the fast Arnoldi method, for a matrix having its eigenvalues distributed over a shifted unit circle. The progressive residual as well as the residual of GMRES are depicted and align nicely. The legend is identical to the one used in Figure 1.

This class of matrices is of the form

$$A = \begin{pmatrix} \Lambda_- & & \\ & \Lambda_+ & \\ & & Z \end{pmatrix}, \tag{6.3}$$

where

$$\Lambda_- = \operatorname{diag}(\lambda_1, \ldots, \lambda_p), \qquad \Lambda_+ = \operatorname{diag}(\lambda_{p+1}, \ldots, \lambda_{n-2}), \qquad Z = \begin{pmatrix} 0 & \gamma \\ -\gamma & 0 \end{pmatrix},$$

with eigenvalues

1. $\lambda_1, \ldots, \lambda_p$ uniformly distributed in the interval $[-\beta, -\alpha]$,
2. $\lambda_{p+1}, \ldots, \lambda_{n-2}$ uniformly distributed in the interval $[\alpha, \beta]$,
3. $\lambda_{n-1}, \lambda_n = \pm\gamma i$.

We take two examples from this class and compare the loss of orthogonality of the recurrence relation (3.6) as predicted in [6] with the Barth-Manteuffel algorithm. The orthogonality of the Arnoldi vectors stored in the matrix V_k is depicted in Figure 5.

As seen in Figure 5 recurrence relation (3.6) gives rise to significantly less orthogonal vectors than the standard Arnoldi iteration. However, it may also be observed that the Barth-Manteuffel algorithm suffers from the same loss of orthogonality as recurrence relation (3.6). We see that the loss of orthogonality emerges as soon as the progressive residual vectors start to differ significantly from the actual GMRES residual, explaining the inaccuracies in the computed vectors. Figure 6 shows the gradual loss of orthogonality between the vectors. White stands for a perfect orthogonality, black for complete loss, The colors assigned are, for 10^0 black (no orthogonality) and for 10^{-16} white (orthogonal up to machine precision). In Figure 6 we observe that v_j is numerically orthogonal to v_k for $j, k <= 14$, $j \neq k$, and also at later stages for $|j-k| \leq 2$, as expected from the local reorthogonalization of our algorithm. However, globally, the orthogonality gets quite quickly lost, as observed already by Embree et al. [6], who suggested Schur complement techniques to tackle this problem.

We can conclude that in this case both Barth-Manteuffel and fast Arnoldi exhibit a fast and almost identical loss of orthogonality.

7 Conclusion

An economic variant for the Arnoldi algorithm has been established for matrices whose adjoint is a low-rank perturbation of a rational function of the matrix. In the process, some aspects

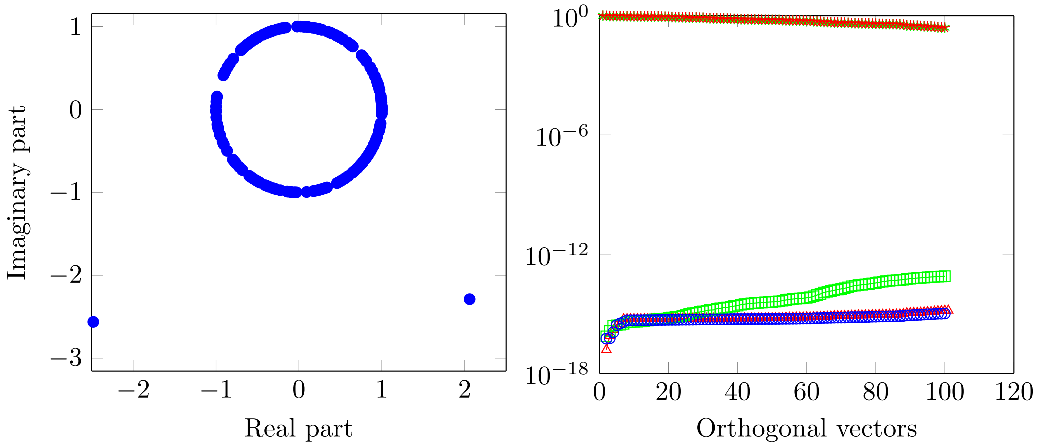

Figure 3: The orthogonality, measured as in [15], of the computed Arnoldi vectors for classical Arnoldi, Barth-Manteuffel, and the fast Arnoldi method, for a matrix having all, except two eigenvalues, on the unit circle. The legend is identical to the one used in Figure 1. The progressive residual as well as the residual of GMRES exhibit the same behavior.

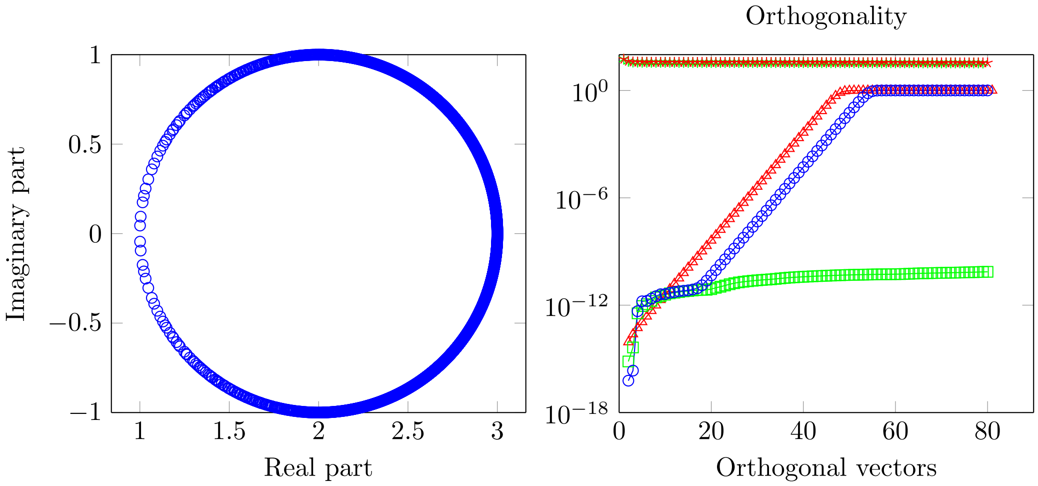

Figure 4: The left plot depicts the eigenvalues of the Neuberger operator $A = \rho I + \gamma\,\mathrm{sign}(Q)$, with $\rho = 2$ and $\gamma = 1$. The right plot shows the accuracy and (progressive) residuals for classical Arnoldi, Barth-Manteuffel, and the fast Arnoldi algorithms. For the legend we refer to Figure 1.

of the Arnoldi process are described in terms of orthogonal polynomials. This includes an explicit formula for the unitary factor in the QR-decomposition of a Hessenberg matrix and a decay property of the entries of this Hessenberg matrix which is related to the convergence of the GMRES algorithm. Also, the existence of a progressive GMRES residual formula has been shown, extending the findings of [4]. Furthermore, comparisons are made with the algorithm described by Barth and Manteuffel [2] for matrices whose adjoint is a low-rank perturbation of a rational function of the matrix, both theoretically and numerically.

8 Acknowledgements

Part of this research has been established during a visit at the university of Science and Technology of Lille in 2014. We thank Bernd Beckermann and Ana Matos for their hospitality.

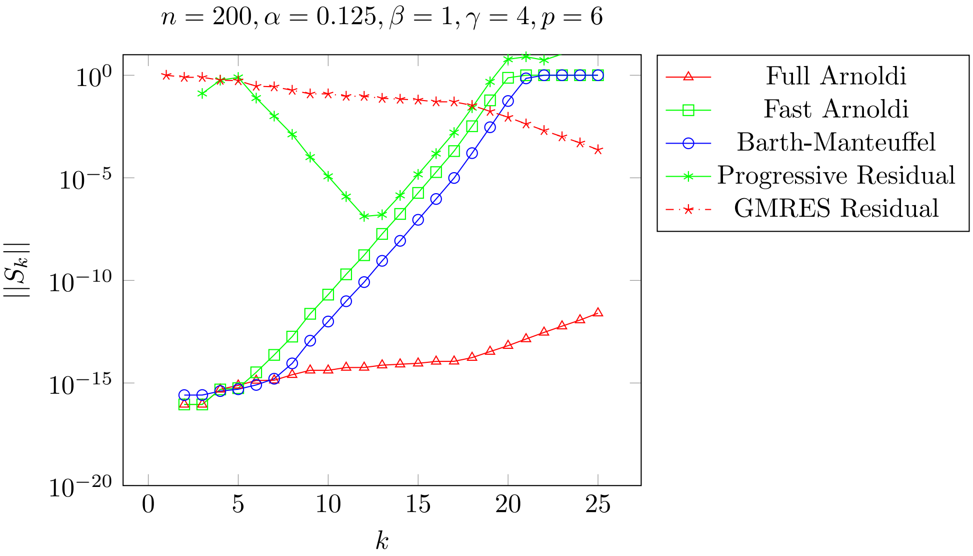

Figure 5: The orthogonality, measured by $\|S_k\|$ as in [15], of the computed Arnoldi vectors for classical Arnoldi, Barth-Manteuffel, and the fast Arnoldi, for a nearly Hermitian matrix of the form (6.3). The progressive residual as well as the residual of GMRES are also depicted, and we see that the loss of accuracy in the fast Arnoldi case is related to inaccurate computations in the progressive residual vectors.

References

[1] G. Arnold, N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, Th. Lippert, and K. Shäfer. *Numerical methods for the QCD overlap operator II: Optimal Krylov subspace methods, in QCD and Numerical Analysis III, eds. A. Boricci, A. Frommer, B. Joo, A. Kennedy and B. Pendleton*, volume 47 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, 2005.

[2] T. Barth and T. Manteuffel. Multiple recursion Conjugate Gradient algorithms Part I: Sufficient conditions. *SIAM Journal on Matrix Analysis and Applications*, 21(3):768–796, 2000.

[3] B. Beckermann. Discrete orthogonal polynomials and superlinear convergence of Krylov subspace methods in numerical linear algebra. *Orthogonal Polynomials and Special Functions, Lecture Notes in Mathematics*, 1883:119–185, 2006.

[4] B. Beckermann and L. Reichel. The Arnoldi process and GMRES for nearly symmetric matrices. *SIAM Journal on Matrix Analysis and Applications*, 30(1):102–120, 2008.

[5] T. Bella, V. Olshevsky, and P. Zhlobich. Classifications of recurrence relations via subclasses of (H,m)-quasiseparable matrices. *Lecture Notes in Electrical Engineering*, 80:23–53, 2011.

[6] M. Embree, J.A. Sifuentes, K.M. Soodhalter, D.B Szyld, and F. Xue. Short-term recurrence Krylov subspace methods for nearly Hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 33(2):480–500, 2012.

[7] V. Faber, J. Liesen, and P. Tichý. The Faber-Manteuffel theorem for linear operators. *SIAM Journal on Numerical Analysis*, 46(3):1323–1337, 2008.

[8] V. Faber and T. Manteuffel. Necessary and sufficient conditions for the existence of a Conjugate Gradient method. *SIAM Journal on Numerical Analysis*, 21(2):352–362, 1984.

[9] Gene Howard Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, USA, third edition, 1996.

[10] W. B. Gragg. The QR algorithm for unitary Hessenberg matrices. *Journal of Computational and Applied Mathematics*, 16:1–8, 1986.

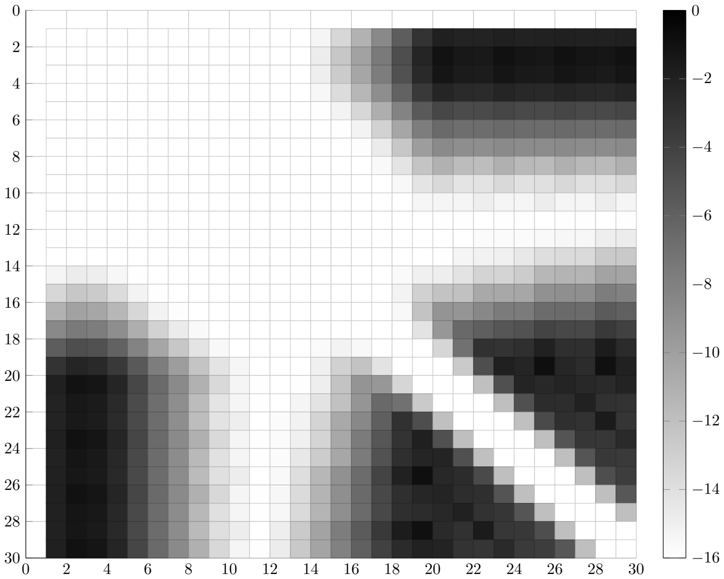

Figure 6: The orthogonality between the successively computed vectors via progressive GMRES is plotted. The picture shows a log scale of the matrix $V_k^* V_k - I$.

[11] C. Jagels and L. Reichel. On the construction of Szegő polynomials. *Journal of Computational and Applied Mathematics*, 46:241–254, 1993.

[12] C. Jagels and L. Reichel. A fast minimal residual algorithm for shifted unitary matrices. *Numerical Linear Algebra with Applications*, 1(6):555–570, 1994.

[13] J. Liesen. When is the adjoint of a matrix a low degree rational function in the matrix? *SIAM Journal on Matrix Analysis and Applications*, 29(4):1171–1180, 2007.

[14] C. Mertens and R. Vandebril. Multiple recurrences and the associated matrix structures stemming from normal matrices. *SIAM Journal on Numerical Analysis*, 52(6):2977–2999, 2014.

[15] C. Paige. A useful form of unitary matrix obtained from any sequence of unit 2-norm n-vectors. *SIAM Journal on Matrix Analysis and Applications*, 31:565–583, 2009.

[16] Y. Saad. *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, PA, USA, 2nd edition, 2003.

[17] J. van den Eshof, A. Frommer, Th. Lippert, K. Shilling, and H.A. van der Vorst. Numerical methods for the QCD overlap operator: I. sign-function and error bounds. *Computational Physics Community*, 146:203–224, 2002.

[18] R. Vandebril, M. Van Barel, and N. Mastronardi. *Matrix Computations and Semiseparable Matrices, Volume I: Linear Systems.* The Johns Hopkins University Press, Baltimore, Maryland, USA, 2008.

```
Data: $A \in \mathbb{C}^{n\times n}$, $\underline{\mathbf{p}}_0$ with $A^* q_m(A) - p_\ell(A) = Q_B(A)$, $p_\ell$ of degree $\ell$, $p_m$ of degree $m$ and $Q_B(A)$ of rank $\kappa$,
    $\{\underline{\phi}_0, \dots, \underline{\phi}_{\kappa-1}\}$ basis of the column space of $Q_B(A)$, $\theta = \begin{cases} \kappa - 1, & \text{if } m > \ell \\ \kappa - \ell + m, & \text{if } m \leq \ell \end{cases}$,
    $\phi = \begin{cases} j - \theta + 1, & \text{if } m > \ell \\ \min\{j - (\ell - m), j - \theta + 1\}, & \text{if } m \leq \ell \end{cases}$ $\underline{c\mathbf{p}}_0 = [\ 1 \quad 0 \quad \dots \quad 0\ ]^T_{m+1}$, $\underline{\rho}_0 = [\ 1 \quad 0 \quad \dots \quad 0\ ]^T_m$
Result: Orthogonal Krylov basis $\underline{\mathbf{p}}_0, \underline{\mathbf{p}}_1, \dots$
for $j = 0, \dots, \theta - 1$ do
    $\underline{\mathbf{p}}_{j+1} = A\underline{\mathbf{p}}_j - \sum_{i=0}^{j} \sigma_{i,j} \underline{\mathbf{p}}_j$,  $\sigma_{i,j} = \frac{\underline{\mathbf{p}}_i^* A \underline{\mathbf{p}}_j}{\underline{\mathbf{p}}_i^* \underline{\mathbf{p}}_i}$ ;
    if $j < m$ then
        $\underline{c\mathbf{p}}_{j+1} = \underline{cA\mathbf{p}}_j - \sum_{i=0}^{j} \sigma_{i,j} \underline{cp}_j$, $\underline{cA\mathbf{p}}_j = [\ 0 \quad \underline{c\mathbf{p}}_j\ ]^T_{m+1}$;
        if $j < m - 1$ then
            $\underline{\rho}_{j+1} = [\ 0 \quad \dots \quad 0 \quad 1 \quad 0 \quad \dots \quad 0\ ]^T_m$;
        else
            solve $\begin{bmatrix} | & | & & | \\ \underline{c\mathbf{p}}_0 & \underline{c\mathbf{p}}_1 & \cdots & \underline{c\mathbf{p}}_m \\ | & | & & | \end{bmatrix} [\ \gamma_0 \quad \dots \quad \gamma_m\ ]^T = [\ b_0 \quad \dots \quad b_m\ ]^T$ for $\underline{\gamma}$, $\underline{\rho}_{j+1} =$
            $\begin{bmatrix} -\frac{\gamma_0}{\gamma_m} & \dots & -\frac{\gamma_{m-1}}{\gamma_m} \end{bmatrix}$;
        end
    else
        $\underline{c}_j = H_{m+1,m} \underline{\rho}_j$,  $\underline{rA\mathbf{p}}_j = c_0 \underline{\rho}_0 + \dots + c_m \underline{\rho}_m$;
        $\underline{\rho}_{j+1} = \underline{rA\mathbf{p}}_j - \sum_{i=0}^{j} \sigma_{i,j} \underline{\rho}_i$;
    end
end
for $j = \theta, \dots,$ do
    solve $[\ \underline{\mu}_j \quad \cdots \quad \underline{\mu}_{j-\kappa+1}\ ]^T \underline{\bar{\tau}}_{j-\theta} = \begin{bmatrix} \underline{\mathbf{p}}^*_{j-\theta} A \underline{\mathbf{p}}_j - \underline{\rho}^*_{j-\theta} \underline{\eta}_j \\ \vdots \\ \underline{\mathbf{p}}^*_{j-\theta} A \underline{\mathbf{p}}_{j-\kappa+1} - \underline{\rho}^*_{j-\theta} \underline{\eta}_{j-\kappa+1} \end{bmatrix}$ for $\underline{\bar{\tau}}_{j-\theta}$;
    $\underline{\mathbf{q}}_{j-\theta_i} = \frac{(\underline{\bar{\rho}}_{j-\theta})_i}{\underline{\mathbf{p}}^*_{j-\theta} \underline{\mathbf{p}}_{j-\theta}} \underline{\mathbf{p}}_{j-\theta} + \underline{\mathbf{q}}_{j-\theta-1_i}$,  $i = 0, \dots, m-1$;
    $\underline{\hat{\mathbf{q}}}_{j-\theta_i} = \frac{(\underline{\bar{\tau}}_{j-\theta})_i}{\underline{\mathbf{p}}^*_{j-\theta} \underline{\mathbf{p}}_{j-\theta}} \underline{\mathbf{p}}_{j-\theta} + \underline{\hat{\mathbf{q}}}_{j-\theta-1_i}$,  $i = 0, \dots, \kappa-1$;
    $\underline{r\mathbf{q}}_{j-\theta_i} = \frac{(\underline{\bar{\rho}}_{j-\theta})_i}{\underline{\mathbf{p}}^*_{j-\theta} \underline{\mathbf{p}}_{j-\theta}} \underline{\rho}_{j-\theta} + \underline{r\mathbf{q}}_{j-\theta-1_i}$,  $i = 0, \dots, m-1$;
    $\underline{r\hat{\mathbf{q}}}_{j-\theta_i} = \frac{(\underline{\bar{\tau}}_{j-\theta})_i}{\underline{\mathbf{p}}^*_{j-\theta} \underline{\mathbf{p}}_{j-\theta}} \underline{\rho}_{j-\theta} + \underline{r\hat{\mathbf{q}}}_{j-\theta-1_i}$,  $i = 0, \dots, \kappa-1$;
    if $j < m$ then
        $\underline{\mathbf{p}}_{j+1} = A\underline{\mathbf{p}}_j - \sum_{i=0}^{j} \sigma_{i,j} \underline{\mathbf{p}}_j$,  $\sigma_{i,j} = \frac{\underline{\mathbf{p}_\mathbf{i}}^* A \underline{\mathbf{p}_\mathbf{j}}}{\underline{\mathbf{p}_\mathbf{i}}^* \underline{\mathbf{p}_\mathbf{i}}}$;
        $\underline{c\mathbf{p}}_{j+1} = \underline{cA\mathbf{p}}_j - \sum_{i=0}^{j} \sigma_{i,j} \underline{cp}_j$, $\underline{cA\mathbf{p}}_j = [\ 0 \quad \underline{c\mathbf{p}}_j\ ]^T_{m+1}$;
        if $j < m - 1$ then
            $\underline{\rho}_{j+1} = [\ 0 \quad \dots \quad 0 \quad 1 \quad 0 \quad \dots \quad 0\ ]^T_m$;
        else
            solve $\begin{bmatrix} | & | & & | \\ \underline{c\mathbf{p}}_0 & \underline{c\mathbf{p}}_1 & \cdots & \underline{c\mathbf{p}}_m \\ | & | & & | \end{bmatrix} [\ \gamma_0 \quad \dots \quad \gamma_m\ ]^T = [\ b_0 \quad \dots \quad b_m\ ]^T$ for $\underline{\gamma}$, $\underline{\rho}_{j+1} =$
            $\begin{bmatrix} -\frac{\gamma_0}{\gamma_m} & \dots & -\frac{\gamma_{m-1}}{\gamma_m} \end{bmatrix}$;
        end
    else
        $\underline{\eta}_j = [\ \underline{\mathbf{p}}_0^* A \underline{\mathbf{p}}_j \quad \cdots \quad \underline{\mathbf{p}}_{m-1}^* A \underline{\mathbf{p}}_j\ ]^T_m$;
        $\underline{\mu}_j = [\ \underline{\phi}_0^* \underline{\mathbf{p}}_j \quad \cdots \quad \underline{\phi}_{\kappa-1}^* \underline{\mathbf{p}}_j\ ]^T_\kappa$;
        $\underline{\mathbf{y}}_{j+1} = A\underline{\mathbf{p}}_j - [\ \underline{\mathbf{q}}_{j-\theta_0} \quad \cdots \quad \underline{\mathbf{q}}_{j-\theta_{m-1}}\ ]\underline{\eta}_j - [\ \underline{\hat{\mathbf{q}}}_{j-\theta_0} \quad \cdots \quad \underline{\hat{\mathbf{q}}}_{j-\theta_{m-1}}\ ]\underline{\mu}_j$;
        $\underline{\mathbf{p}}_{j+1} = \underline{\mathbf{y}}_{j+1} - \sum_{i=\phi}^{j} \hat{t}_{i,j} \underline{\mathbf{p}}_i$,  $\hat{t}_{i,j} = \frac{\underline{\mathbf{p}}_i^* \underline{\mathbf{y}}_{j+1}}{\underline{\mathbf{p}}_i^* \underline{\mathbf{p}}_i}$;
        $\underline{c}_j = H_{m+1,m} \underline{\rho}_j$,  $\underline{rA\mathbf{p}}_j = c_0 \underline{\rho}_0 + \dots + c_m \underline{\rho}_m$;
        $\underline{r\mathbf{y}}_{j+1} = \underline{rA\mathbf{p}}_j - \sum_{i=0}^{m-1} (\underline{\eta}_j)_i \underline{r\mathbf{q}}_{j-\theta_i} - \sum_{i=0}^{\kappa-1} (\underline{\mu}_j)_i \underline{r\hat{\mathbf{q}}}_{j-\theta_i}$;
        $\underline{\rho}_{j+1} = \underline{r\mathbf{y}}_{j+1} - \sum_{i=\phi}^{j} \hat{t}_{i,j} \underline{\rho}_i$;
    end
end
```

Algorithm 2: Barth-Manteuffel algorithm