CAM 1246

Quasi-kernel polynomials and their use in non-Hermitian matrix iterations *

Roland W. Freund
AT&T Bell Laboratories, Murray Hill, NJ 07974, United States

Received 5 November 1991
Revised 11 February 1992

Abstract

Freund, R.W., Quasi-kernel polynomials and their use in non-Hermitian matrix iterations, Journal of Computational and Applied Mathematics 43 (1992) 135–158.

Some of the most efficient iterative algorithms for large sparse Hermitian matrix computations are based on orthogonal or kernel polynomials. For the case of non-Hermitian matrices, methods based on orthogonal or kernel polynomials are less satisfactory, in that the resulting algorithms involve long recurrences. Consequently, it is usually too expensive to run the full algorithms and restarts are necessary. A typical example is the generalized minimal residual method (GMRES) for solving non-Hermitian linear systems, where work and storage per iteration grow linearly with the iteration number. Recently, two quasi-minimal residual methods (QMR) for solving non-Hermitian linear systems have been proposed, which — unlike GMRES — are based on short recurrences and hence can be used as true iterative schemes, without restarts. In this paper, the concept of quasi-kernel polynomials is introduced. Some general theory for quasi-kernel polynomials is developed, such as recurrence relations and a characterization of roots of quasi-kernel polynomials as generalized eigenvalues. It is pointed out that the QMR approaches are based on two particular instances of quasi-kernel polynomials. Also, the use of quasi-kernel polynomials for approximating eigenvalues or pseudospectra of large sparse non-Hermitian matrices is briefly discussed.

Keywords: Non-Hermitian matrices; matrix iterations; orthogonal polynomials; kernel polynomials; generalized minimal residual method; quasi-kernel polynomials; recurrence relation; roots of quasi-kernel polynomials; quasi-minimal residual algorithm; eigenvalue approximations.

1. Introduction

In the 1950s, Stiefel wrote his classical paper [26] on orthogonal and kernel polynomials and their use in iterative computations for Hermitian matrices. Today, two of the methods

Correspondence to: Dr. R.W. Freund, AT&T Bell Laboratories, Room 2C-420, 600 Mountain Avenue, Murray Hill, NJ 07974-0636, United States.

* Work supported by the Defense Advanced Research Projects Agency via Cooperative Agreement NCC 2-387 between the National Aeronautics and Space Administration and the Universities Space Research Association.

discussed in [26], namely the conjugate gradient algorithm (CG) due to Hestenes and Stiefel [13] for solving Hermitian positive definite linear systems and the symmetric Lanczos process [14] for the Hermitian eigenvalue problem, are recognized as very powerful tools for large sparse Hermitian matrix computations. An especially attractive feature of both methods is that they can be implemented using only short vector recursions, which just reflects the three-term recurrences for orthogonal and kernel polynomials corresponding to inner products on the real line.

Since the early 1980s, there has been considerable interest in developing generalizations of CG and the symmetric Lanczos process for non-Hermitian matrix computations. Such extensions can again be based on orthogonal or kernel polynomials, but now corresponding to inner products in the complex plane. Examples of methods of this type are the generalized minimal residual algorithm (GMRES) due to Saad and Schultz [22] for solving non-Hermitian linear systems and the Arnoldi process [1,19] for the non-Hermitian eigenvalue problem. However, in general these methods now involve long vector recursions, which reflects the fact that complex orthogonal and kernel polynomials only satisfy long recurrences. Consequently, it is usually too expensive to run the full version of these algorithms and restarts are necessary, which often leads to very slow convergence.

For this reason, recent research in non-Hermitian matrix iterations has focused mainly on schemes that are based on short recurrences. In particular, there has been a revival of the nonsymmetric Lanczos process, and it is now well understood how the possible breakdowns and potential instabilities in the classical algorithm [14] can be overcome by using look-ahead techniques (see [6,11,18] and the references given therein). Based on the look-ahead Lanczos method, Freund and Nachtigal [8] have developed a novel CG-type approach for solving general non-Hermitian linear systems, the quasi-minimal residual algorithm (QMR). The QMR scheme produces iterates that are characterized by a quasi-minimization of the residual norm, rather than a true residual minimization property as in GMRES. The point is that the QMR iterates — unlike the GMRES iterates — can be computed using only short recurrences and hence QMR can be used as a true iterative scheme, without restarts. Furthermore, the quasi-minimization property is powerful enough to ensure that QMR converges smoothly, with convergence rates comparable to those of full GMRES without restarts (see [7,8] for theoretical and numerical results).

The QMR iteration is based on the nonsymmetric Lanczos process and, like the latter, requires matrix-vector multiplications with the coefficient matrix A, as well as its transpose A^{T}. This is a disadvantage for some applications, where A^{T} is not readily available. In [5], we developed a variant of QMR, the transpose-free QMR algorithm, which does not require matrix-vector multiplications with A^{T}.

The GMRES method is closely connected with classical kernel polynomials in the complex plane. This naturally leads to the question about the corresponding polynomials for QMR and its transpose-free variant.

In this paper, an answer to this question is given. First, we introduce the general concept of quasi-kernel polynomials, and then we point out that QMR and transpose-free QMR are based on two particular instances of quasi-kernel polynomials. Moreover, we develop some theory for general quasi-kernel polynomials, such as recurrence relations and a characterization of roots of quasi-kernel polynomials as generalized eigenvalues. As a by-product, we obtain a new algorithm for computing zeros of standard kernel polynomials. Finally, we discuss the use of quasi-kernel polynomials for approximating eigenvalues or pseudospectra of large sparse non-Hermitian matrices.

The remainder of this paper is organized as follows. In Section 2, we briefly review residual and standard kernel polynomials. In Section 3, we introduce the concept of quasi-kernel polynomials. In Section 4, we derive a recurrence relation. In Section 5, it is shown how zeros of kernel and quasi-kernel polynomials can be computed by solving a generalized eigenvalue problem. In Section 6, we turn to GMRES, QMR and transpose-free QMR and consider the kernel and quasi-kernel polynomials on which these algorithms are based. In Section 7 some numerical examples are presented. Finally, in Section 8, we make some concluding remarks.

Throughout the paper, all vectors and matrices, unless otherwise stated, are assumed to be complex. As usual, $M^{\mathrm{T}}=[m_{kj}]$ and $M^{\mathrm{H}}=[\overline{m_{kj}}]$ denote the transpose and the conjugate transpose, respectively, of the matrix $M=[m_{jk}]$. We use the symbol I_n for the $n\times n$ identity matrix. The vector norm $\|x\|_2=\sqrt{x^{\mathrm{H}}x}$ is always the Euclidean norm. The set of eigenvalues of a square matrix M is denoted by $\lambda(M)$, and if M is Hermitian, $\lambda_{\max}(M)$ is the largest eigenvalue of M. We use the notation

$$\mathscr{P}_n := \{\varphi(\lambda)\equiv\sigma_0+\sigma_1\lambda+\cdots+\sigma_n\lambda^n \mid \sigma_0,\sigma_1,\ldots,\sigma_n\in\mathbb{C}\}$$

for the set of all complex polynomials of degree at most n. The set of all complex polynomials is $\mathscr{P}_\infty$, and deg φ denotes the degree of $\varphi\in\mathscr{P}_\infty$. The symbol 0 will be used for the number zero, the zero matrix, and the polynomial $\varphi(\lambda)\equiv 0$; its actual meaning and, in the case of the zero matrix, its dimension will be apparent from the context.

2. Residual and kernel polynomials

Kernel polynomials arise naturally in iterative matrix computations. For example, consider linear systems

$$Ax=b, \tag{2.1}$$

where A is a large sparse nonsingular, in general complex, $N\times N$ matrix. Many iterative schemes for solving (2.1), such as GMRES and QMR, belong to the class of *Krylov subspace methods*: they produce approximations x_n to $A^{-1}b$ of the form

$$x_n\in x_0+K_n(r_0,A),\quad n=1,2,\ldots\,. \tag{2.2}$$

Here $x_0\in\mathbb{C}^N$ is any initial guess for the solution of (2.1), $r_0:=b-Ax_0$ is the corresponding residual vector, and

$$K_n(r_0,A):=\operatorname{span}\{r_0,Ar_0,\ldots,A^{n-1}r_0\} \tag{2.3}$$

is the nth *Krylov subspace* generated by r_0 and A. In view of

$$K_n(r_0,A)=\{\varphi(A)r_0\mid\varphi\in\mathscr{P}_{n-1}\}, \tag{2.4}$$

the nth iterate (2.2) can be expressed in terms of polynomials:

$$x_n=x_0+\rho_n(A)r_0,\quad\text{where } \rho_n\in\mathscr{P}_{n-1}. \tag{2.5}$$

Therefore, schemes with iterates (2.2) are also referred to as *polynomial acceleration methods*.

We remark that the residual vector corresponding to the iterate (2.5) is given by

$$r_n := b - Ax_n = \psi_n(A)r_0, \tag{2.6}$$

where ρ_n and ψ_n are connected as follows:

$$\psi_n(\lambda) \equiv 1 - \lambda\rho_n(\lambda). \tag{2.7}$$

Note that

$$\psi_n \in \mathscr{P}_n \quad \text{and} \quad \psi_n(0) = 1. \tag{2.8}$$

Generally, any ψ_n satisfying (2.8) is called an nth *residual polynomial.*

Clearly, the goal for the design of polynomial acceleration methods is to choose ψ_n in (2.6) such that r_n is as small as possible, subject to the constraint (2.8). A standard approach is to require that the Euclidean norm of the residual is minimal, i.e.,

$$\| r_n \|_2 = \min_{\psi \in \mathscr{P}_n:\ \psi(0) = 1} \| \psi(A) r_0 \|_2. \tag{2.9}$$

GMRES is an algorithm that computes iterates defined by the minimization property (2.9).

Setting $\lambda_0 := 0$ and

$$\langle \varphi, \psi \rangle := r_0^{\mathrm{H}} (\psi(A))^{\mathrm{H}} \varphi(A) r_0, \tag{2.10}$$

we see that (2.9) is an instance of a polynomial approximation problem of the more general class

$$\min_{\psi \in \mathscr{P}_n:\ \psi(\lambda_0) = 1} \langle \psi, \psi \rangle. \tag{2.11}$$

Here $\lambda_0 \in \mathbb{C}$ is any fixed number and $\langle \cdot, \cdot \rangle$ is a given positive definite inner product on $\mathscr{P}_n$. We remark that other strategies (see [21,24]) for choosing the residual polynomial ψ_n in (2.6) also lead to problems of the form (2.11), e.g., with inner products given by

$$\langle \varphi, \psi \rangle := \int\int_G \varphi(\lambda)\overline{\psi(\lambda)}\omega(\lambda)\, \mathrm{d}\sigma \quad \text{or} \quad \langle \varphi, \psi \rangle := \int_C \varphi(\lambda)\overline{\psi(\lambda)}\omega(\lambda)\, |\mathrm{d}\lambda|. \tag{2.12}$$

Typically, $G \subset \mathbb{C}$ is a compact set containing the spectrum $\lambda(A)$ of A or some approximation to $\lambda(A)$, and C is a curve bounding such a set G. Moreover, $\omega > 0$ is some suitably chosen weight function on G or C, respectively. Finally, we remark that approximation problems (2.11) also arise in the context of iterative algorithms for computing a few eigenvalues of large sparse matrices A, where polynomials defined via (2.11) are used to dampen components of the starting vector along the unwanted parts of $\lambda(A)$ (see [20]).

It is well known (see [2, Chapter I], [26], [27, Chapter XVI] or Corollary 3.3 below) that the solution of (2.11) can be expressed in terms of kernel polynomials. More precisely, let $\varphi_j \in \mathscr{P}_j$, $j = 0, 1, \ldots, n$, be a set of *orthonormal polynomials* with respect to $\langle \cdot, \cdot \rangle$, i.e.,

$$\langle \varphi_j, \varphi_k \rangle = \begin{cases} 0, & \text{if } j \neq k, \\ 1, & \text{if } j = k, \end{cases} \tag{2.13}$$

and set

$$\chi_n(\lambda_0, \lambda) \equiv \sum_{j=0}^{n} \overline{\varphi_j(\lambda_0)}\varphi_j(\lambda). \tag{2.14}$$

Then, the normalized polynomial

$$\psi_n(\lambda) \equiv \frac{\chi_n(\lambda_0, \lambda)}{\chi_n(\lambda_0, \lambda_0)} \tag{2.15}$$

is the unique solution of the approximation problem (2.11). Note that the function $\chi_n(\lambda_0, \lambda)$ is called nth *kernel polynomial*, since for each fixed $\lambda_0 \in \mathbb{C}$ it satisfies

$$\langle \varphi(\cdot), \chi_n(\lambda_0, \cdot)\rangle = \varphi(\lambda_0) \quad \text{for all } \varphi \in \mathscr{P}_n.$$

For the discussion of kernel and quasi-kernel polynomials one may assume, without loss of generality, that $\lambda_0 = 0$ is fixed. Clearly, the general case $\lambda_0 \in \mathbb{C}$ can be reduced to this special case by means of the linear transformation $\lambda \mapsto \lambda - \lambda_0$. Therefore, from now on, we only consider the case $\lambda_0 = 0$. Finally, in the sequel, the notation ψ_n is used to denote an nth kernel or quasi-kernel polynomial corresponding to $\lambda_0 = 0$ and normalized as in (2.15), i.e., $\psi_n(0) = 1$. Note that, in view of (2.8), ψ_n is always a residual polynomial.

3. Quasi-kernel polynomials

In this section, we introduce the concept of quasi-kernel polynomials. We use the following general setting.

3.1. The setting

From now on, it is always assumed that $\langle \cdot, \cdot \rangle$ is a given positive semidefinite inner product on $\mathscr{P}_\infty$, i.e., for all $\varphi_1, \varphi_2, \varphi, \psi \in \mathscr{P}_\infty$ and $\sigma_1, \sigma_2 \in \mathbb{C}$:

$$\begin{gathered}\langle \sigma_1\varphi_1 + \sigma_2\varphi_2, \psi\rangle = \sigma_1\langle \varphi_1, \psi\rangle + \sigma_2\langle \varphi_2, \psi\rangle, \\ \overline{\langle \varphi, \psi\rangle} = \langle \psi, \varphi\rangle, \qquad \langle \varphi, \varphi\rangle \geqslant 0.\end{gathered} \tag{3.1}$$

Notice that we do not require $\langle \varphi, \varphi\rangle > 0$ for $\varphi \neq 0$, i.e., $\langle \cdot, \cdot \rangle$ is not assumed to be positive definite. Clearly, the inner products defined in (2.10) and (2.12) all fulfil the conditions (3.1). Furthermore, let $\Pi_J := \{\varphi_j\}_{j=0}^J$, where J can either be a finite integer or equal to ∞, be a given sequence of polynomials

$$\varphi_j \in \mathscr{P}_j, \quad \text{with } \deg \varphi_j = j, \tag{3.2}$$

which span $\mathscr{P}_J$. Note that φ_0 is a nonzero constant polynomial.

We will use the notation $\Phi_n(\lambda)$, where $\lambda \in \mathbb{C}$, for the row vector

$$\Phi_n(\lambda) := [\varphi_0(\lambda) \quad \varphi_1(\lambda) \quad \cdots \quad \varphi_n(\lambda)]. \tag{3.3}$$

In the sequel, it is always assumed that $n \in \{1, 2,\ldots, J\}$, if J is finite, respectively $n \in \{1, 2,\ldots\}$, if $J = \infty$.

Notice that, by (3.2), each polynomial $\lambda\varphi_j$ can be represented as a linear combination of $\varphi_0, \varphi_1,\ldots, \varphi_{j+1}$. Therefore, we have the identity

$$\lambda\Phi_{n-1}(\lambda) \equiv \Phi_n(\lambda)H_n, \tag{3.4}$$

where

$$H_n = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & h_{n,n-1} & h_{nn} \\ 0 & \cdots & 0 & h_{n+1,n} \end{bmatrix} \tag{3.5}$$

is an $(n+1) \times n$ upper Hessenberg matrix. Furthermore, all subdiagonal elements of H_n are nonzero, i.e.,

$$h_{j+1,j} \neq 0, \quad j = 1, 2, \ldots, n, \tag{3.6}$$

and this ensures that

$$\operatorname{rank} H_n = n. \tag{3.7}$$

We remark that, from (3.4) (with $\lambda = 0$), one obtains

$$\Phi_n(0) H_n = 0, \quad \text{where } \Phi_n(0) \neq 0; \tag{3.8}$$

here the last inequality is guaranteed by $\varphi_0(0) = \varphi_0 \neq 0$.

3.2. The concept of quasi-kernel polynomials

If the polynomials in Π_j satisfy the orthogonality relations (2.13), then the kernel polynomials corresponding to $\langle \cdot, \cdot \rangle$ are simply given by (2.14) and (2.15). Unfortunately, for inner products in the complex plane, such as (2.10) and (2.12), orthogonal polynomials do not satisfy short recurrences in general (cf. [15]). As a consequence, for iterative matrix computations based on an orthogonal basis Π_j, work per iteration and storage requirements grow linearly with the iteration index n, and typically, it is not practical to use such schemes for large n. However, useful nonorthogonal bases Π_j can often be generated cheaply, e.g., using simple three-term recurrences (cf. Section 6). Roughly speaking, quasi-kernel polynomials are approximations to the true kernel polynomials, which are derived from a given general basis Π_j, rather than an orthogonal basis.

Next, we turn to the exact definition of quasi-kernel polynomials. Obviously, in view of (3.2), one has the parametrization

$$\psi_n(\lambda) \equiv 1 - \lambda \Phi_{n-1}(\lambda) z, \quad z \in \mathbb{C}^n, \tag{3.9}$$

for all possible residual polynomials (2.8). Using (3.4), we rewrite (3.9) in the form

$$\psi_n(\lambda) \equiv \Phi_n(\lambda)(d_n - H_n z), \quad \text{where } d_n := \begin{bmatrix} 1/\varphi_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{C}^{n+1}, \tag{3.10}$$

which implies

$$\langle \psi_n, \psi_n \rangle = (d_n - H_n z)^{\mathrm{H}} G_n (d_n - H_n z). \tag{3.11}$$

Here

$$G_n := \left[\langle \varphi_j, \varphi_k \rangle\right]_{j,k=0,1,\ldots,n}$$

is the Gram matrix of $\varphi_0, \varphi_1, \ldots, \varphi_n$. Note that G_n is Hermitian positive semidefinite, and from (3.11) one obtains the estimate

$$\langle \psi_n, \psi_n \rangle \leqslant \lambda_{\max}(G_n) \| d_n - H_n z \|_2^2. \tag{3.12}$$

Recall that, in (3.9), $z \in \mathbb{C}^n$ is still a free parameter vector. If we choose z such that the right-hand side in (3.11) is minimal, then ψ_n is just the true kernel polynomial. However, this would require knowledge of the matrix G_n, which is not available in typical applications (cf. Section 6). Therefore, instead of (3.11), we only minimize the second factor in the upper bound (3.12). The precise definition is as follows.

Definition 3.1. Let $n \geqslant 1$ and let $z_n \in \mathbb{C}^n$ be the solution of the least-squares problem

$$\| d_n - H_n z_n \|_2 = \min_{z \in \mathbb{C}^n} \| d_n - H_n z \|_2. \tag{3.13}$$

The polynomial ψ_n given by (3.9) (with $z := z_n$) is called the nth *quasi-kernel polynomial* (corresponding to the inner product $\langle \cdot , \cdot \rangle$ and derived from Π_j). For $n = 0$, we set $\psi_0(\lambda) \equiv 1$.

Note that, by (3.7), the matrix H_n has full column rank, and therefore the least-squares problem (3.13) has exactly one solution. Hence, ψ_n is always uniquely determined by Definition 3.1.

For the special case thrt the basis polynomials in Π_j are orthonormal (see (2.13)), we have $G_n = I_{n+1}$, and (3.11) reduces to

$$\langle \psi_n, \psi_n \rangle = \| d_n - H_n z_n \|_2^2.$$

Therefore, for orthonormal bases Π_j, quasi-kernel polynomials are identical with standard kernel polynomials. Here and in the sequel, we use the adjectives *true* and *standard* to distinguish kernel polynomials from quasi-kernel polynomials.

Finally, we remark that often (cf. Section 6) the polynomials in Π_j are chosen to have unit norm, i.e.,

$$\langle \varphi_j, \varphi_j \rangle = 1 \quad \text{for all } j. \tag{3.14}$$

From (3.14), by the Cauchy–Schwarz inequality, one obtains the estimates

$$|\langle \varphi_j, \varphi_k \rangle| \leqslant 1 \quad \text{for all } j, k,$$

which readily imply that the first factor in (3.12) is bounded as follows:

$$\lambda_{\max}(G_n) \leqslant n + 1. \tag{3.15}$$

3.3. A formula for quasi-kernel polynomials

Next, we show that quasi-kernel polynomials can be represented directly in terms of the basis polynomials in Π_j.

Tneorem 3.2. *The nth quasi-kernel polynomial ψ_n is given by*

$$\psi_n(\lambda) \equiv \frac{\sum_{j=0}^{n} \overline{\varphi_j(0)}\varphi_j(\lambda)}{\sum_{j=0}^{n} |\varphi_j(0)|^2}. \tag{3.16}$$

Proof. From Definition 3.1 and (3.10), we have

$$\psi_n(\lambda) \equiv \Phi_n(\lambda)y_n, \quad \text{where } y_n := d_n - H_n z_n = [d_n \quad H_n]\begin{bmatrix} 1 \\ -z_n \end{bmatrix}. \tag{3.17}$$

Furthermore, as solution of (3.13), z_n satisfies the normal equations corresponding to (3.13), and thus

$$H_n^{\mathrm{H}} y_n = 0. \tag{3.18}$$

Note that, in view of (3.5), (3.6) and (3.10), the matrix $[d_n \ H_n]$ in (3.17) is upper triangular and nonsingular, and this guarantees $y_n \neq 0$. On the other hand, by (3.8), we have

$$H_n^{\mathrm{H}} \Phi_n(0)^{\mathrm{H}} = 0, \quad \text{where } \Phi_n(0)^{\mathrm{H}} \neq 0. \tag{3.19}$$

By (3.7), the $n \times (n+1)$ matrix H_n^{H} has a one-dimensional null space. Hence, from (3.18) and (3.19), it follows that

$$y_n = \frac{1}{\sigma}\Phi_n(0)^{\mathrm{H}}, \quad \text{where } \sigma \in \mathbb{C},\ \sigma \neq 0. \tag{3.20}$$

Finally, by inserting (3.20) in (3.17) and using the fact that $\psi_n(0) = 1$, we obtain the identity

$$\psi_n(\lambda) \equiv \frac{\Phi_n(\lambda)\Phi_n(0)^{\mathrm{H}}}{\sigma}, \quad \text{where } \sigma = \Phi_n(0)\Phi_n(0)^{\mathrm{H}}, \tag{3.21}$$

which is just (3.16). □

For the special case of orthonormal bases Π_j, Theorem 3.2 reduces to the following well-known property (cf. Section 2) of standard kernel polynomials.

Corollary 3.3. *Assume that the polynomials Π_j satisfy the orthonormality relations* (2.13). *Then the polynomial ψ_n given by* (3.16) *is the unique optimal solution of the approximation problem* (2.11). *In particular, ψ_n is the nth kernel polynomial corresponding to $\lambda_0 = 0$ and normalized such that $\psi_n(0) = 1$.*

4. A recurrence relation

For practical purposes, the representation (3.16) is not very useful, since in general all previous basis polynomials are required. In this section, we show that ψ_n can always be updated by means of a short recursion, which involves only ψ_n, ψ_{n-1} and φ_n.

Our starting point is the least-squares problem (3.13). We use the standard approach (see, e.g., [9, Chapter 5]) to solving (3.13) based on a QR factorization of H_n:

$$Q_n H_n = \begin{bmatrix} R_n \\ 0 \end{bmatrix}, \tag{4.1}$$

where Q_n is unitary and R_n is a nonsingular upper triangular matrix. Since H_n is an upper Hessenberg matrix, the matrix Q_n can be chosen as a product of n Givens rotations, i.e.,

$$Q_n = G_n \begin{bmatrix} G_{n-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} G_{n-2} & 0 \\ 0 & I_2 \end{bmatrix} \cdots \begin{bmatrix} G_1 & 0 \\ 0 & I_{n-1} \end{bmatrix}, \tag{4.2}$$

where, for $j = 1, 2, \ldots, n$,

$$G_j = \begin{bmatrix} I_{j-1} & 0 & 0 \\ 0 & c_j & -\bar{s}_j \\ 0 & s_j & c_j \end{bmatrix}, \quad \text{with } c_j \in \mathbb{R},\ s_j \in \mathbb{C},\ c_j^2 + |s_j|^2 = 1. \tag{4.3}$$

Furthermore, Q_n and the matrix Q_{n-1} obtained from the QR factorization of the previous Hessenberg matrix H_{n-1} differ only in the nth Givens rotation:

$$Q_n = G_n \begin{bmatrix} Q_{n-1} & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.4}$$

Using (4.1) and the fact that Q_n is unitary, we obtain the identity

$$\| d_n - H_n z \|_2 = \left\| Q_n d_n - \begin{bmatrix} R_n \\ 0 \end{bmatrix} z \right\|_2 \quad \text{for all } z \in \mathbb{C}^n,$$

from which it follows that

$$z_n = R_n^{-1} t_n, \quad \text{where } t_n = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix} \in \mathbb{C}^n, \quad \begin{bmatrix} t_n \\ \tilde{\tau}_{n+1} \end{bmatrix} := Q_n d_n, \tag{4.5}$$

is the solution of (3.13). Moreover, we have

$$Q_n(d_n - H_n z_n) = \tilde{\tau}_{n+1} e_{n+1} \quad \text{and} \quad \| d_n - H_n z_n \|_2 = |\tilde{\tau}_{n+1}|, \tag{4.6}$$

where $e_{n+1} := [0 \ \cdots \ 0 \ 1]^{\mathrm{T}} \in \mathbb{R}^{n+1}$. With (4.4) and (4.3) one easily verifies that τ_n and $\tilde{\tau}_{n+1}$ in (4.5) can be updated as follows:

$$\tau_n = c_n \tilde{\tau}_n \quad \text{and} \quad \tilde{\tau}_{n+1} = s_n \tilde{\tau}_n, \qquad \text{where, if } n = 1,\ \tilde{\tau}_1 := \frac{1}{\varphi_0}. \tag{4.7}$$

Finally, by combining (3.10) (with $z = z_n$) and the first relation in (4.6), one gets

$$\psi_n(\lambda) \equiv \tilde{\tau}_{n+1} \Phi_n(\lambda) Q_n^{\mathrm{H}} e_{n+1}. \tag{4.8}$$

Using (4.8), (3.3), (4.4), (4.3) (for $j = n$), the second relation in (4.7) and again (4.8) (with n replaced by $n - 1$), we obtain

$$\begin{aligned} \psi_n(\lambda) &\equiv \tilde{\tau}_{n+1} [\Phi_{n-1}(\lambda) Q_{n-1}^{\mathrm{H}} \quad \varphi_n(\lambda)] G_n^{\mathrm{H}} e_{n+1} \\ &\equiv \tilde{\tau}_{n+1} \bar{s}_n \Phi_{n-1}(\lambda) Q_{n-1}^{\mathrm{H}} e_n + c_n \tilde{\tau}_{n+1} \varphi_n(\lambda) \\ &\equiv |s_n|^2 \psi_{n-1}(\lambda) + c_n \tilde{\tau}_{n+1} \varphi_n(\lambda). \end{aligned}$$

The last identity is the desired recurrence relation. Notice that, in view of (4.7), $\bar{\tau}_{n+1}$ can be expressed in terms of the parameters s_j of the Givens rotations:

$$\bar{\tau}_{n+1} = \frac{1}{\varphi_0} s_1 s_2 \cdots s_n.$$

Hence we have proved the following result.

Theorem 4.1. *For $n = 1, 2, \ldots, J$, if J is finite, respectively $n = 1, 2, \ldots$, if $J = \infty$, it holds*

$$\psi_n = |s_n|^2 \psi_{n-1} + c_n \bar{\tau}_{n+1} \varphi_n, \quad \bar{\tau}_{n+1} = \frac{1}{\varphi_0} s_1 s_2 \cdots s_n. \tag{4.9}$$

We remark that, as a by-product of our derivation, we obtain the estimate

$$\langle \psi_n, \psi_n \rangle \leqslant \lambda_{\max}(G_n) \left| \frac{1}{\varphi_0} s_1 s_2 \cdots s_n \right|^2,$$

which follows from (3.12) and the second relations in (4.6) and (4.9).

Recall from Section 2 that, for polynomial acceleration methods, the iterates (2.5) involve the polynomials ρ_n defined by (2.7). Next, we give an update formula for these polynomials.

Theorem 4.2. *For $n = 1, 2, \ldots, J$, if J is finite, respectively $n = 1, 2, \ldots$, if $J = \infty$, it holds*

$$\rho_n = \rho_{n-1} + \tau_n \zeta_{n-1}, \quad \tau_n = \frac{1}{\varphi_0} s_1 s_2 \cdots s_{n-1} c_n, \tag{4.10}$$

where ζ_{n-1} is defined by

$$\Sigma_{n-1}(\lambda) := [\zeta_0(\lambda) \quad \zeta_1(\lambda) \quad \cdots \quad \zeta_{n-1}(\lambda)] := \Phi_{n-1}(\lambda) R_n^{-1}. \tag{4.11}$$

Proof. Using (2.7), (3.9) (with $z = z_n$), (4.5) and (4.11), we obtain that

$$\rho_n(\lambda) \equiv \Phi_{n-1}(\lambda) z_n \equiv \Phi_{n-1}(\lambda) R_n^{-1} t_n \equiv \Sigma_{n-1}(\lambda) t_n. \tag{4.12}$$

The first relation in (4.10) is then an immediate consequence of (4.12) (considered for n and for $n-1$), (4.5) and (4.11). The formula for τ_n in (4.10) follows from (4.7). □

Theorems 4.1 and 4.2 show that quasi-kernel polynomials ψ_n and the corresponding polynomials ρ_n can be generated by means of short recurrences, provided that the basis polynomials φ_j, respectively the polynomials ζ_j, satisfy short recursions. Note that, in view of (3.4), the polynomials φ_j can be generated by s-term recurrence relations if, and only if, the matrix $H_n = [h_{jk}]$ is banded with upper bandwidth $s-2$, i.e., $h_{jk} = 0$ if $k > j + s - 2$. Using (4.2) and (4.3), one readily verifies that the upper triangular matrix R_n in (4.1) is banded with s diagonals, if H_n is banded with upper bandwidth $s-2$. Consequently, by (4.11), the polynomials ζ_j can be generated by s-term recurrence relations, if the polynomials φ_j satisfy s-term recursions. For example, consider the important special case $s = 3$, i.e., the matrix H_n is tridiagonal and the basis polynomials φ_j fulfil three-term recurrences. Here the matrix R_n is of the form

$$R_n = \begin{bmatrix} \rho_1 & \xi_2 & \theta_3 & 0 & \cdots & 0 \\ 0 & \rho_2 & \xi_3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \rho_3 & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \theta_n \\ \vdots & & & \ddots & \ddots & \xi_n \\ 0 & \cdots & \cdots & \cdots & 0 & \rho_n \end{bmatrix}, \quad \text{where } \rho_j \neq 0, \ j = 1, 2, \ldots, n, \tag{4.13}$$

and the polynomials ζ_j satisfv the recurrences

$$\zeta_{j+1} = \frac{\varphi_{j+1} - \xi_j \zeta_j - \theta_j \zeta_{j-1}}{\rho_j}.$$

5. Zeros of quasi-kernel polynomials

In this section, we consider the problem of computing zeros of quasi-kernel polynomials. In particular, it is shown that zeros of quasi-kernel polynomials can be characterized as generalized eigenvalues. Recall from Section 3 that kernel polynomials are a special case of quasi-kernel polynomials. It appears that the results in this section are new even for standard kernel polynomials.

5.1. Preliminaries

Throughout this section, n is assumed to be fixed. We denote by $\lambda_1, \lambda_2, \ldots, \lambda_L \in \mathbb{C} \cup \{\infty\}$ the distinct zeros (possibly including ∞) of ψ_n, and m_l is the multiplicity of λ_l, $l = 1, 2, \ldots, L$. Since $\psi_n(0) = 1$, we have the representation

$$\psi_n(\lambda) \equiv \prod_{l=1}^{L} \left(1 - \frac{\lambda}{\lambda_l}\right)^{m_l} \tag{5.1}$$

of ψ_n. Here an infinite zero $\lambda_l = \infty$ is included if, and only if, ψ_n does not have full degree n, and we set $m_l := n - \deg \psi_n$ then. From (3.16), one immediately obtains the following characterization of infinite zeros.

Proposition 5.1. *The following conditions are equivalent*:

(i) ψ_n *has an infinite zero* $\lambda_l = \infty$ *of multiplicity* m_l;

(ii) $\varphi_n(0) = \varphi_{n-1}(0) = \cdots = \varphi_{n-m_l+1}(0) = 0$ *and* $\varphi_{n-m_l}(0) \neq 0$;

(iii) $\psi_n = \psi_{n-1} = \cdots = \psi_{n-m_l+1} = \psi_{n-m_l} \neq \psi_{n-m_l-1}$.

For the derivation of Theorem 5.2 below, an infinite zero would require a special treatment. To avoid this, it will be more convenient to work with the reversed polynomial

$$\hat{\psi}_n(\mu) \equiv \mu^n \psi_n\left(\frac{1}{\mu}\right), \tag{5.2}$$

instead of ψ_n. Note that $\hat{\psi}_n$ always has finite eigenvalues $\mu_l := 1/\lambda_l \in \mathbb{C}$, $l = 1, 2, \ldots, L$, and, by (5.1) and (5.2), it holds

$$\hat{\psi}_n(\mu) \equiv \prod_{l=1}^{L} (\mu - \mu_l)^{m_l}.$$

We will also need transformed versions of the row vectors (3.3), and we set, for all $\mu \in \mathbb{C}$,

$$\hat{\Phi}_n(\mu) := \mu^n \Phi_n\left(\frac{1}{\mu}\right) \quad \text{and} \quad \hat{\Phi}_{n-1}(\mu) := \mu^{n-1} \Phi_{n-1}\left(\frac{1}{\mu}\right). \tag{5.3}$$

Notice that $\hat{\Phi}_n(\mu)$ and $\hat{\Phi}_{n-1}(\mu)$ are row vectors of length $n+1$ and n, respectively, and they are connected by the relation

$$\hat{\Phi}_n(\mu) \begin{bmatrix} I_n \\ 0 \end{bmatrix} \equiv \mu \hat{\Phi}_{n-1}(\mu). \tag{5.4}$$

In terms of (5.3), the relation (3.4) now reads

$$\hat{\Phi}_{n-1}(\mu) \equiv \hat{\Phi}_n(\mu) H_n. \tag{5.5}$$

Furthermore, with (3.21) and (5.3), we obtain the representation

$$\hat{\psi}_n(\mu) \equiv \frac{\hat{\Phi}_n(\mu)\Phi_n(0)^{\mathrm{H}}}{\Phi_n(0)\Phi_n(0)^{\mathrm{H}}} \tag{5.6}$$

of the reversed quasi-kernel polynomial (5.2).

By (3.3) and (5.3), the entries of $\hat{\Phi}_n(\mu)$ and $\hat{\Phi}_{n-1}(\mu)$ are polynomials in μ and hence analytic. Therefore, all derivatives

$$\hat{\Phi}_n^{(k)}(\mu) := \frac{\mathrm{d}^k}{\mathrm{d}\mu^k} \hat{\Phi}_n(\mu) \quad \text{and} \quad \hat{\Phi}_{n-1}^{(k)}(\mu) := \frac{\mathrm{d}^k}{\mathrm{d}\mu^k} \hat{\Phi}_{n-1}(\mu), \qquad k = 0, 1, \ldots,$$

exist. We remark that the condition (3.2) guarantees that

$$\hat{\Phi}_n^{(k)}(\mu) \neq 0 \quad \text{for all } \mu \in \mathbb{C} \text{ and } k = 0, 1, \ldots, n. \tag{5.7}$$

By differentiating (5.4) k times and applying Leibniz's rule for the differentiation of the product on the right-hand side of (5.4), one readily verifies that

$$\hat{\Phi}_n^{(k)}(\mu) \begin{bmatrix} I_n \\ 0 \end{bmatrix} \equiv \mu \hat{\Phi}_{n-1}^{(k)}(\mu) + k \hat{\Phi}_{n-1}^{(k-1)}(\mu), \quad k = 0, 1, \ldots . \tag{5.8}$$

Moreover, by (5.5), it holds

$$\hat{\Phi}_{n-1}^{(k)}(\mu) \equiv \hat{\Phi}_n^{(k)}(\mu) H_n, \quad k = 0, 1, \ldots . \tag{5.9}$$

Finally, in the sequel, we will denote by

$$\tilde{H}_n := [I_n \quad 0] H_n \tag{5.10}$$

the $n \times n$ matrix obtained from H_n by deleting the last row in (3.5). Also, we will make use of the identity

$$\begin{aligned}\mathscr{R}(H_n) &:= \{H_n a \mid a \in \mathbb{C}^n\}\\ &= \{y \in \mathbb{C}^{n+1} \mid \Phi_n(0) y = 0\} =: \mathscr{N}(\Phi_n(0)),\end{aligned} \tag{5.11}$$

which follows from (3.7) and (3.8).

5.2. Connection with a generalized eigenvalue problem

After these preliminaries, we now prove the main result of this section.

Theorem 5.2. *The zeros μ_l, $l = 1, 2, \ldots, L$, of the polynomial $\hat{\psi}_n$ are the eigenvalues of*

$$C_n := \tilde{H}_n^{\mathrm{H}} (H_n^{\mathrm{H}} H_n)^{-1}.$$

Moreover, each μ_l has algebraic multiplicity m_l and unit geometric multiplicity, and the vectors $a_k^{(l)} \neq 0$, $k = 0, 1, \ldots, m_l - 1$, which are uniquely defined by

$$H_n a_k^{(l)} = \frac{1}{k!} \left(\hat{\Phi}_n^{(k)}(\mu_l)\right)^{\mathrm{H}}, \quad a_k^{(l)} \in \mathbb{C}^n, \tag{5.12}$$

are corresponding left eigenvectors, respectively left principal vectors:

$$\begin{aligned}&(a_0^{(l)})^{\mathrm{H}} (C_n - \mu_l I_n) = 0,\\ &(a_k^{(l)})^{\mathrm{H}} (C_n - \mu_l I_n) = (a_{k-1}^{(l)})^{\mathrm{H}}, \quad k = 1, 2, \ldots, m_l - 1.\end{aligned} \tag{5.13}$$

Proof. Let $l \in \{1, 2, \ldots, L\}$ be arbitrary, but fixed. For simplicity, we suppress indices l and set $\mu := \mu_l$, $m := m_l$ and $a_k := a_k^{(l)}$. Thus μ is a zero of $\hat{\psi}_n$ of multiplicity m, i.e.,

$$\hat{\psi}_n^{(k)}(\mu) = 0, \quad k = 0, 1, \ldots, m - 1, \qquad \hat{\psi}_n^{(m)}(\mu) \neq 0. \tag{5.14}$$

In the sequel, it is always assumed that $k \in \{0, 1, \ldots, m - 1\}$.

From (5.14), (5.6) and (5.7), we obtain the identity

$$\Phi_n(0) \left(\hat{\Phi}_n^{(k)}(\mu)\right)^{\mathrm{H}} = 0, \quad \text{where } \left(\hat{\Phi}_n(\mu)\right)^{\mathrm{H}} \neq 0,$$

which, since $\mathscr{R}(H_n) = \mathscr{N}(\Phi_n(0))$ (cf. (5.11)), implies that there exists a vector $a_k \neq 0$ satisfying (5.12). Furthermore, by (3.7), H_n has full column rank, and therefore a_k is uniquely determined by (5.12).

It remains to show that a_k fulfils (5.13). With (5.12) and (5.9), it follows that

$$k!\, a_k^{\mathrm{H}} H_n^{\mathrm{H}} H_n = \hat{\Phi}_n^{(k)}(\mu) H_n = \hat{\Phi}_{n-1}^{(k)}(\mu), \tag{5.15}$$

and, if $k \geqslant 1$,

$$(k - 1)!\, a_{k-1}^{\mathrm{H}} H_n^{\mathrm{H}} H_n = \hat{\Phi}_{n-1}^{(k-1)}(\mu). \tag{5.16}$$

By using (5.10), (5.12), (5.8), (5.15) and (5.16), we obtain

$$\begin{aligned} k!\,a_k^{\mathrm H}\tilde{H}_n^{\mathrm H} &= k!\,a_k^{\mathrm H}H_n^{\mathrm H}\begin{bmatrix} I_n \\ 0 \end{bmatrix} = \hat{\Phi}_n^{(k)}(\mu)\begin{bmatrix} I_n \\ 0 \end{bmatrix} \\ &= \mu\hat{\Phi}_{n-1}^{(k)}(\mu) + k\hat{\Phi}_{n-1}^{(k-1)}(\mu) \\ &= k!\,\mu a_k^{\mathrm H}H_n^{\mathrm H}H_n + \begin{cases} 0, & \text{if } k=0, \\ k!\,a_{k-1}^{\mathrm H}H_n^{\mathrm H}H_n, & \text{if } k \geqslant 1. \end{cases} \end{aligned} \tag{5.17}$$

Finally, by multiplying (5.17) from the right by $(1/k!)(H_n^{\mathrm H}H_n)^{-1}$, one arrives at (5.13). □

By rewriting the main result of Theorem 5.2 for the original quasi-kernel polynomial (5.1), the following corollary results.

Corollary 5.3. *The nth quasi-kernel polynomial satisfies*

$$\psi_n(\lambda) \equiv \frac{\det\left(H_n^{\mathrm H}H_n - \lambda\tilde{H}_n^{\mathrm H}\right)}{\det\left(H_n^{\mathrm H}H_n\right)}.$$

In particular, the zeros of ψ_n are the eigenvalues of the generalized eigenvalue problem

$$H_n^{\mathrm H}H_n z = \lambda\tilde{H}_n^{\mathrm H}z, \quad z \in \mathbb{C}^n, \ z \neq 0. \tag{5.18}$$

5.3. Computing zeros of quasi-kernel polynomials

In view of Corollary 5.3, zeros of quasi-kernel polynomials can be computed by solving a generalized eigenvalue problem. The standard method for this task is the QZ algorithm [16]. However, since $\operatorname{cond}_2(H_n^{\mathrm H}H_n) = (\operatorname{cond}_2(H_n))^2$ (here we denote by $\operatorname{cond}_2(M) := \max_{x:\|x\|_2=1}\|Mx\|_2/\min_{x:\|x\|_2=1}\|Mx\|_2$ the Euclidean condition number of a matrix M), the matrix $H_n^{\mathrm H}H_n$ can be ill-conditioned, even if the condition number of H_n is moderate, and thus in general it is not advisable to solve the generalized eigenvalue problem in the form (5.18). Next, we show how this problem can be avoided by rewriting (5.18) by means of the QR decomposition (4.1), which we have already used in Section 4.

Let Q_n and R_n be the unitary and upper triangular factors in (4.1), where again we assume that Q_n is a product (4.2) of n Givens rotations (4.3). Then, by (4.1), it holds

$$R_n^{-\mathrm H}H_n^{\mathrm H}H_n = R_n, \tag{5.19}$$

and, by (5.10) and (4.1), we have

$$R_n^{-\mathrm H}\tilde{H}_n^{\mathrm H} = R_n^{-\mathrm H}H_n^{\mathrm H}\begin{bmatrix} I_n \\ 0 \end{bmatrix} = \begin{bmatrix} I_n & 0 \end{bmatrix} Q_n \begin{bmatrix} I_n \\ 0 \end{bmatrix} =: \tilde{Q}_n. \tag{5.20}$$

Note that, by (4.2) and (4.3) (for $j = n$),

$$\tilde{Q}_n = \begin{bmatrix} I_{n-1} & 0 \\ 0 & c_n \end{bmatrix} G_{n-1} \begin{bmatrix} G_{n-2} & 0 \\ 0 & 1 \end{bmatrix} \cdots \begin{bmatrix} G_1 & 0 \\ 0 & I_{n-2} \end{bmatrix}. \tag{5.21}$$

Finally, by multiplying (5.18) from the left by R_n^{-H} and by using (5.19) and (5.20), we obtain the following equivalent formulation of (5.18):

$$R_n z = \lambda \tilde{Q}_n z, \quad z \in \mathbb{C}^n, \ z \neq 0. \tag{5.22}$$

Therefore, the roots of quasi-kernel polynomials can be computed as follows.

Algorithm 5.4 (For computing the zeros of ψ_n).

(1) Compute the QR factorization (4.1) of H_n with Q_n of the form (4.2), (4.3);

(2) Compute $\tilde{Q}_n$ using (5.21);

(3) Apply the QZ algorithm to the generalized eigenvalue problem (5.22). The resulting eigenvalues are the zeros of ψ_n.

Since R_n is nonsingular, $\lambda = \infty$ is an eigenvalue of (5.22) if, and only if, $\tilde{Q}_n$ is singular. In view of (5.21), it follows that ψ_n has an infinite zero if, and only if, $c_n = 0$.

We would like to point out that Algorithm 5.4 seems to be new even for the special case of standard kernel polynomials. Saylor and Smolarski [23] have proposed an algorithm for computing roots of kernel polynomials, which is based on a formulation of the problem as a standard eigenvalue problem. However, this formulation is only possible if ψ_n has full degree n, and the case deg $\psi_n < n$ is treated in [23] by applying the algorithm to $\psi_{n'}$, where $n' = \deg \psi_n$ (cf. Proposition 5.1). The problem with this approach is that it requires the numerical determination of the degree of ψ_n, which can be tricky if ψ_n has leading coefficients close to zero. Note that in Algorithm 5.4 this decision is avoided by always working with a generalized eigenvalue problem.

6. GMRES, QMR and transpose-free QMR

In this section, we return to linear systems (2.1), and we consider the algorithms GMRES, QMR and transpose-free QMR for solving (2.1).

All three methods can be formulated using the setting from Section 3, and their iterates can be expressed via kernel or quasi-kernel polynomials. In all cases, $\langle \cdot, \cdot \rangle$ is the inner product defined in (2.10), and in the sequel, we denote by

$$\|\varphi\|_{\langle\rangle} := \langle \varphi, \varphi \rangle^{1/2}, \quad \varphi \in \mathcal{P}_\infty,$$

the corresponding seminorm. Note that $\langle \cdot, \cdot \rangle$ is positive definite on $\mathcal{P}_{J_A - 1}$, i.e.,

$$\langle \varphi, \varphi \rangle > 0 \quad \text{for all } \varphi \in \mathcal{P}_{J_A - 1}, \ \varphi \neq 0,$$

where

$$J_A := \dim K_N(r_0, A) \tag{6.1}$$

denotes the degree of the minimal polynomial of r_0 with respect to A (cf. [29, p.37]).

All three algorithms generate a sequence of basis vectors $v_0, v_1, \ldots, v_J$ for the Krylov subspaces (2.3):

$$K_n(r_0, A) = \text{span}\{v_0, v_1, \ldots, v_{n-1}\}, \quad n = 1, 2, \ldots, J.$$

In view of (2.4), these basis vectors are of the form

$$v_j = \varphi_j(A)r_0, \quad \text{where } \varphi_j \in \mathscr{P}_j, \ \deg \varphi_j = j, \tag{6.2}$$

and, as in Section 3, we denote by $\Pi_J = \{\varphi_j\}_{j=0}^J$ the set of these basis polynomials. Then, the iterates produced by all three methods are given by

$$x_n = x_0 + [v_0 \quad v_1 \quad \cdots \quad v_{n-1}]z_n,$$

where z_n is defined in (3.13), and the corresponding residual vectors are

$$r_n = \psi_n(A)r_0, \tag{6.3}$$

where ψ_n is the nth quasi-kernel polynomial (corresponding to the inner product $\langle \cdot, \cdot \rangle$ and derived from Π_J). We remark that, by (2.10), (6.2) and (6.3), it holds

$$\| v_j \|_2 = \| \varphi_j \|_{\langle\rangle}, \ j = 0, 1, \ldots, J, \quad \text{and} \quad \| r_n \|_2 = \| \psi_n \|_{\langle\rangle}, \ n = 0, 1, \ldots, J. \tag{6.4}$$

The three algorithms GMRES, QMR and transpose-free QMR differ in the choice of the set of basis polynomials Π_J. Next we discuss the three different cases.

6.1. GMRES

Here $J = J_A$ (cf. (6.1)), $\varphi_0, \varphi_1, \ldots, \varphi_{J-1}$ are orthonormal with respect to the positive definite inner product (2.10), and φ_J is a minimal polynomial of r_0 with respect to A. In view of Corollary 3.3, the residual polynomials ψ_n in (6.3) are true kernel polynomials corresponding to (2.10), and the GMRES iterates are characterized by the minimal residual property (2.9). Moreover, in exact arithmetic, the full GMRES algorithm stops after J iterations with the solution $x_J = A^{-1}b$ of (2.1), and all roots of the last kernel polynomial ψ_J are eigenvalues of A.

In the actual GMRES algorithm, the vectors (6.2) are constructed by means of the Arnoldi process [1], based on the recurrences (3.4) for the polynomials in Π_J. Unfortunately, except for very special cases [4], the matrix H_n in (3.4) is a full upper Hessenberg matrix, and consequently, the construction of a new basis vector v_n in general involves all previous basis vectors $v_0, v_1, \ldots, v_{n-1}$. In particular, work and storage per iteration grow linearly with the iteration number n. Therefore, in practice, it becomes prohibitive to run the full algorithm for a large number of iterations, and usually GMRES is used with restarts.

We would like to point out that the linear growth of work and storage requirements per iteration is not a consequence of the particular implementation used in GMRES, but — as was shown in [4] — it is indeed true for any method that generates iterates defined by the minimal residual property (2.9).

6.2. QMR

In the QMR approach, the nonsymmetric Lanczos process is used to generate the polynomials in Π_J.

The nonsymmetric Lanczos method is based on the bilinear form

$$[\varphi, \psi] := s_0^{\mathrm{T}}\psi(A)\varphi(A)r_0, \quad \varphi, \psi \in \mathscr{P}_\infty. \tag{6.5}$$

Here $s_0 \in \mathbb{C}^N$, $s_0 \neq 0$, is a second starting vector, which can be chosen freely; usually, one sets $s_0 = \overline{r_0}$ or chooses s_0 as a vector with random entries. The Lanczos process generates polynomials $\phi_0, \phi_1, \ldots, \phi_J$ that are orthogonal with respect to (6.5):

$$[\phi_j, \phi_k] = 0, \quad \text{if } j \neq k. \tag{6.6}$$

We remark that, for the Hermitian case, i.e., $A = A^{\mathrm{H}}$ and $s_0 = \overline{r_0}$, the Lanczos polynomials ϕ_j are real [26]. Therefore, in the Hermitian case, it suffices to consider the bilinear form (6.5) for real polynomials only, and then (6.5) and the inner product (2.10) are identical. However, in the general non-Hermitian case, the bilinear form is not a positive semidefinite inner product (cf. (3.1)). In particular, there exist polynomials $\varphi \neq 0$ with negative "length" $[\varphi, \varphi] < 0$.

Generally, polynomials that satisfy the orthogonality relations (6.6) for a given bilinear form $[\cdot, \cdot]$ are called *formally orthogonal polynomials* (FOPs) (see [3,11] and the references given therein). Unlike true orthogonal polynomials in the complex plane, FOPs can always be generated by means of short recursions; usually, they even satisfy three-term recurrences of the type

$$\gamma_{j+1}\phi_j(\lambda) \equiv (\lambda - \alpha_j)\phi_{j-1}(\lambda) - \beta_j\phi_{j-2}(\lambda), \quad \gamma_{j+1}, \alpha_j, \beta_j \in \mathbb{C}, \ \gamma_{j+1} \neq 0. \tag{6.7}$$

However, if one insists on three-term recurrences (6.7) in each step, then exact or near-breakdowns of the process cannot be excluded. Fortunately, such an event is extremely unlikely in practice; moreover, if it occurs, it can be overcome by resorting to slightly longer recursions and relaxing the orthogonality conditions (6.6). For details, we refer the reader to [6,11,12].

In the QMR algorithm [8], the Lanczos polynomials

$$\varphi_j := \phi_j, \quad j = 0, 1, \ldots, J,$$

corresponding to the bilinear form (6.6) are chosen as the basis polynomials in Π_J. Here $J := J_{\mathrm{L}}$ where J_{L} is defined as the largest integer such that there exists a unique monic polynomial φ of degree J_{L} with $[\varphi, \psi] = 0$ for all $\psi \in \mathscr{P}_{J_{\mathrm{L}}-1}$. We remark that J is the termination index of the Lanczos method (see [6]) and that $J_{\mathrm{L}} \leqslant J_A$.

For the actual construction of the vectors (6.2), we use the implementation of the Lanczos process that was recently developed in [6]. In this particular variant of the Lanczos method, exact or near-breakdowns are handled by means of so-called *look-ahead* techniques (see [6,18]). If all the Lanczos polynomials can be generated by three-term recurrences (6.7), then the matrix H_n in (3.4) is a tridiagonal matrix; this is the generic case. If look-ahead steps of size bigger than 1 are taken due to exact or near-breakdowns, then H_n still remains upper Hessenberg and also exhibits a block tridiagonal structure:

$$H_n = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \gamma_2 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \gamma_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_{l(n)} \\ \vdots & & \ddots & \gamma_{l(n)} & \alpha_{l(n)} \\ 0 & \cdots & 0 & 0 & \gamma_{l(n)+1} \end{bmatrix}. \tag{6.8}$$

Here $l(n)$ is the number of steps in the look-ahead Lanczos algorithm, and the diagonal blocks α_k are small square matrices whose size corresponds to the length of the kth look-ahead step.

In particular, $l(n)=n$ and all α_k's are 1×1 blocks if no exact or near-breakdowns were encountered.

The block tridiagonal structure (6.8) of H_n is the reason why QMR can be implemented using short recurrences only. Indeed, with (6.8), (3.4) and (3.3), it follows that the basis vectors (6.2) can be generated by means of short recursions. In the generic case they even satisfy three-term recurrences. Notice that, in view of Theorem 4.1, the nth QMR residual vector (6.3) is always a linear combination of r_{n-1} and v_n. Updating the corresponding iterate x_n is slightly more complicated, but it can also be computed by means of a short recursion (see [8] for details), which again reduces to a three-term recurrence in the generic case.

We remark that, in the QMR algorithm, the vectors v_j are normalized to have unit length. Therefore, in view of (6.4), it holds

$$\langle \varphi_j, \varphi_j \rangle = \| v_j \|_2^2 = 1 \quad \text{for all } j,$$

and thus the condition (3.14) is satisfied. Then, with (3.12), (3.13), (3.15) and the second relation in (6.4), we obtain the following upper bound for the norm of the QMR residual vector r_n:

$$\| r_n \|_2 \leqslant \sqrt{n+1} \min_{z\in\mathbb{C}^n} \| d_n - H_n z \|_2. \tag{6.9}$$

Based on the estimate (6.9), it is possible to prove a convergence theorem for QMR (see [8, Theorem 6.1]).

6.3. Transpose-free QMR

In addition to the basis vectors (6.2), the standard QMR method based on the look-ahead Lanczos process also involves a second sequence of vectors given by

$$w_j = \phi_j(A^{\mathrm{T}})s_0, \quad j=0,1,\ldots,J_{\mathrm{L}}.$$

Generating these vectors requires matrix-vector multiplications with A^{T}, which is a disadvantage for certain applications where A^{T} is not readily available. On the other hand, in the QMR algorithm, the vectors w_j are only used in the computations of vector products of the form

$$w^{\mathrm{T}}v = [\varphi, \psi], \quad \text{where } v=\varphi(A)r_0,\ w=\psi(A^{\mathrm{T}})s_0,\ \varphi, \psi \in \mathscr{P}_{J_{\mathrm{L}}-1}. \tag{6.10}$$

Now, by means of (6.5), the product in (6.10) can be rewritten as follows:

$$w^{\mathrm{T}}v = s_0^{\mathrm{T}}(\psi(A)\varphi(A))r_0 = [1, \psi\varphi]. \tag{6.11}$$

Therefore, if we choose any set of basis polynomials Π_J where $J \geqslant 2(J_{\mathrm{L}}-1)$, then all possible products (6.10) can be obtained — without using A^{T} — as linear combinations of

$$s_0^{\mathrm{T}}v_j = s_0^{\mathrm{T}}\varphi_j(A)r_0, \quad j=0,1,\ldots,J.$$

We remark that the crucial relation (6.11) was first used in [25] in the derivation of the transpose-free conjugate gradients squared algorithm.

For the transpose-free QMR method (TFQMR), which we proposed in [5], the basic polynomials in Π_J, where $J := 2J_{\mathrm{L}} - 1$, are chosen as products of the Lanczos polynomials ϕ_k,

$k=0,\ 1,\ldots,J_{\rm L}$, defined by (6.6). More precisely, the TFQMR algorithm is based on the polynomials

$$\varphi_j=\begin{cases}\dfrac{\phi_k^2}{\|\phi_k^2\|_{\langle\,\rangle}}, & \text{if } j=2k, \quad k=0,\ 1,\ldots,J_{\rm L}-1,\\[2ex] \dfrac{\phi_{k-1}\phi_k}{\|\phi_{k-1}\phi_k\|_{\langle\,\rangle}}, & \text{if } j=2k-1, \quad k=1,\ 2,\ldots,J_{\rm L}-1,\\[2ex] \phi_{J_{\rm L}-1}\phi_{J_{\rm L}}, & \text{if } j=2J_{\rm L}-1.\end{cases} \tag{6.12}$$

We remark that, using the fact that $J_{\rm L}\leqslant J_A$, one can easily check that the denominators in (6.12) are always positive.

Note that the polynomials (6.12), except for the last one $\varphi_{2J_{\rm L}-1}$, are normalized such that (3.14) holds. Consequently, for TFQMR, we again have an estimate of the type (6.9), where, however, H_n is now the recurrence matrix corresponding to the polynomials (6.12), rather than the Lanczos matrix (6.8). In particular, we would like to stress that the iterates produced by TFQMR and standard QMR are different in general.

For details of an actual implementation of TFQMR, we refer the reader to [5].

7. Numerical experiments

In this section, we present two numerical examples.

7.1. Example 1

This example is a nonsymmetric linear system (2.1). We consider the partial differential equation

$$Lu=f \quad \text{on } (0,\ 1)\times(0,\ 1)\times(0,\ 1), \tag{7.1}$$

where

$$\begin{aligned}Lu={}&-\frac{\partial}{\partial x}\left(\mathrm{e}^{xy}\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(\mathrm{e}^{xy}\frac{\partial u}{\partial y}\right)-\frac{\partial}{\partial z}\left(\mathrm{e}^{xy}\frac{\partial u}{\partial z}\right)\\ &+\beta(x+y+z)\frac{\partial u}{\partial x}+\left(\gamma+\frac{1}{1+x+y+z}\right)u,\end{aligned}$$

with Dirichlet boundary conditions $u=0$. The right-hand side f is chosen such that

$$u=(1-x)(1-y)(1-z)(1-\mathrm{e}^{-x})(1-\mathrm{e}^{-y})(1-\mathrm{e}^{-z})$$

is the exact solution of (7.1). We set the parameters in (7.1) to $\beta=30$ and $\gamma=-250$, and then we discretize (7.1) using centered differences on a uniform $15\times15\times15$ grid with mesh size $h=\frac{1}{16}$. This leads to a linear system (2.1) with a sparse nonsymmetric coefficient matrix A of order $N=3375$. We have solved this system with GMRES, QMR and TFQMR. In all three cases, we chose x_0 as starting vector, and as stopping criterion we used

$$\frac{\|r_n\|_2}{\|r_0\|_2}\leqslant 10^{-8}. \tag{7.2}$$

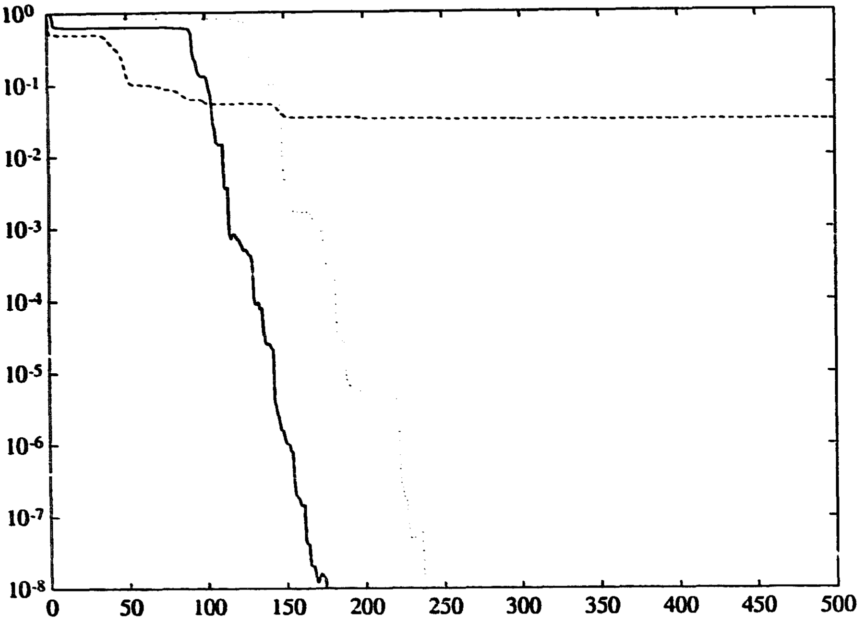

Fig. 7.1. Relative residual norms $\|r_n\|_2 / \|r_0\|_2$ plotted versus n, for Example 1.

For the second starting vector s_0 in QMR and TFQMR, a vector with random entries was chosen. To avoid excessive storage requirements, the GMRES algorithm was restarted after each cycle of 50 iterations. In Fig. 7.1, we have plotted the relative residual norm (7.2) versus the iteration number n. The solid line is the convergence curve for standard QMR, the dotted line shows the behavior of TFQMR, and the dashed line is the GMRES convergence curve. We remark that, in terms of work and storage requirements, one iteration of QMR corresponds to two iterations of TFQMR. Therefore, for this example TFQMR is the best method. Note that GMRES — due to its minimal residual property — is optimal before it is restarted for the first time after 50 steps; after that it begins to stagnate.

7.2. Example 2

In this example, we compute roots of the kernel and quasi-kernel polynomials associated with GMRES and QMR, respectively, using Algorithm 5.4.

Both GMRES and QMR were applied to the linear system (2.1), where A is a 200×200 Toeplitz matrix of the form

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & \cdots & 0 \\ -1 & 1 & 1 & 1 & 1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 1 & 1 \\ \vdots & & \ddots & \ddots & \ddots & 1 & 1 \\ \vdots & & & \ddots & -1 & 1 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & -1 & 1 \end{bmatrix}. \qquad (7.3)$$

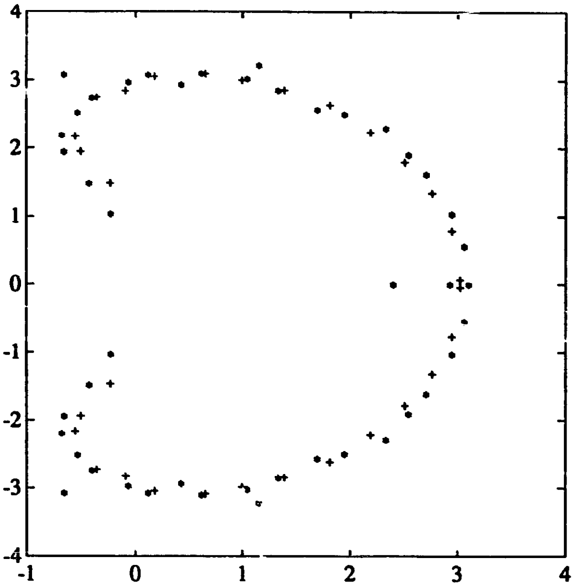

Fig. 7.2. Zeros of GMRES and QMR residual polynomials, for Example 2.

We remark that the matrix (7.3) was first introduced in [10] as a test example for iterative algorithms. The starting residual r_0 and, for QMR, the second starting vector s_0 were both chosen as random vectors. In Fig. 7.2, we show the zeros (marked by +) of the kernel polynomial $\psi_{30}^{\mathrm{GMRES}}$ corresponding to the GMRES residual r_{30}^{GMRES} after 30 steps, as well as the zeros (marked by $*$) of the quasi-kernel polynomial ψ_{50}^{QMR} corresponding to the QMR residual r_{50}^{QMR} after 50 steps.

Recall from Section 6 that, in exact arithmetic, GMRES will terminate after a finite number of steps J, and all roots of ψ_J^{GMRES} are eigenvalues of A. For $n < J$, the zeros of ψ_n^{GMRES} can be considered as approximations to the eigenvalues of A. Figure 7.2 shows that the approximate eigenvalues obtained from GMRES and QMR exhibit a similar convergence behavior. Since QMR is based only on a quasi-minimization property, the convergence for QMR is slower than for GMRES. Hence, in Fig. 7.2, we compare the roots of ψ_{50}^{QMR} with those of $\psi_{30}^{\mathrm{GMRES}}$.

The matrix A also appears in [17] as an example to demonstrate that, for nonnormal matrices A, the convergence behavior of Krylov subspace methods depends on the pseudospectrum [28] of A, rather than the spectrum $\lambda(A)$. Unfortunately, pseudospectra for large matrices A cannot be computed directly. Instead, it was pointed out in [17] that the sets bounded by lemniscates

$$C_n(\eta) := \{\lambda \in \mathbb{C} \mid |\psi_n(\lambda)| = \eta\}$$

of GMRES polynomials usually are good approximations to pseudospectra. Of course, for practical applications, one needs to pick a value for the parameter η; a heuristic for this choice can be found in [17]. The GMRES polynomials are generated by computing their coefficients explicitly in [17], which is an unstable procedure for polynomials of high degree, and then the

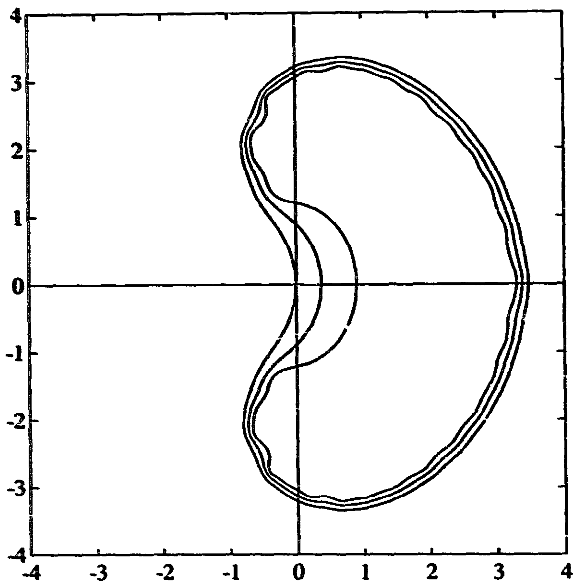

Fig. 7.3. GMRES lemniscates, for Example 2.

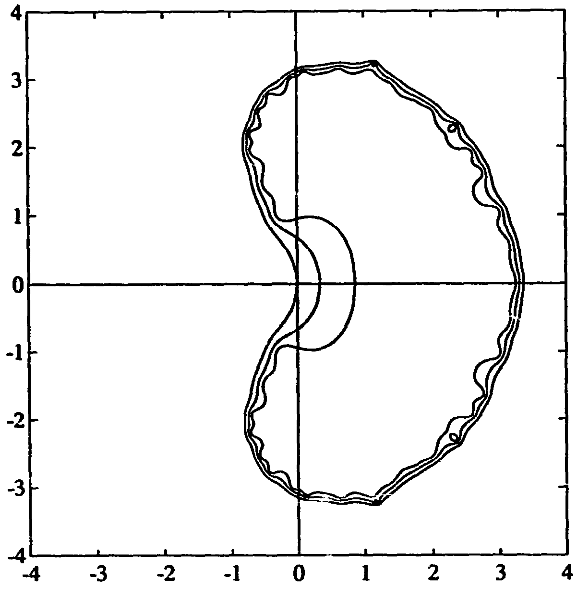

Fig. 7.4. QMR lemniscates, for Example 2.

roots are obtained. By using Algorithm 5.4, roots of general quasi-kernel polynomials can be computed directly and in a numerically stable manner.

In Fig. 7.3, the lemniscates

$$C_n(\eta) \quad \text{for } \eta = \mathrm{e}^{-t},\ t = 0, 1, 2, \tag{7.4}$$

of the GMRES polynomial $\psi_{30}^{\mathrm{GMRES}}$ are plotted. In Fig. 7.4, we show the corresponding lemniscates (7.4) for the QMR polynomial ψ_{50}^{QMR}. In both cases, the polynomials were obtained from their roots, which were computed by means of Algorithm 5.4. As can be seen from the plots, both approaches give roughly the same approximations to the pseudospectrum.

8. Concluding remarks

In this paper, we have introduced the concept of quasi-kernel polynomials. Roughly speaking, quasi-kernel polynomials are approximations to true kernel polynomials obtained from a seı of arbitrary basis polynomials, rather than orthonormal polynomials. We have presented some results for general quasi-kernel polynomials, such as recurrence relations and a characterization of roots of quasi-kernel polynomials as generalized eigenvalues. If the basis polynomials satisfy short recurrences, then the corresponding quasi-kernel polynomials can also be generated by means of short recursions. As a result, matrix iterations based on such quasi-kernel polynomials can be implemented with short vector updates. In contrast, true kernel polynomials usually do not fulfil short recursions, and consequently, iterative schemes based on true kernel polynomials generally involve long vector recurrences.

We have shown that two recently proposed quasi-minimal residual algorithms for solving non-Hermitian linear systems are based on particular instances of quasi-kernel polynomials.

Finally, the use of quasi-kernel polynomials for approximating eigenvalues and pseudospectra of non-Hermitian matrices was discussed briefly.

Acknowledgements

I would like to thank Susanne Freund, Marlis Hochbruck and Noël Nachtigal for their careful reading of the manuscript and for their suggestions, which helped to improve the presentation of this paper. Noël Nachtigal also provided Matlab routines for Example 2 in Section 7. I would like to thank the referees for their constructive comments.

References

[1] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, *Quart. Appl. Math.* **9** (1951) 17–29.

[2] T.S. Chihara, *An Introduction to Orthogonal Polynomials* (Gordon and Breach, New York, 1978).

[3] A. Draux, *Polynômes Orthogonaux Formels — Applications*, Lecture Notes in Math. **974** (Springer, Berlin, 1983).

[4] V. Faber and T. Manteuffel, Necessary and sufficient conditions for the existence of a conjugate gradient method, *SIAM J. Numer. Anal.* **21** (1984) 352–362.

[5] R.W. Freund, A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, *SIAM J. Sci. Statist. Comput.*, to appear.

[6] R.W. Freund, M.H. Gutknecht and N.M. Nachtigal, An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, *SIAM J. Sci. Statist. Comput.*, to appear.

[7] R.W. Freund and N.M. Nachtigal, An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, Part II, RIACS Technical Report 90.46, NASA Ames Research Center, 1990.

[8] R.W. Freund and N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, *Numer. Math.* **60** (1991) 315–339.

[9] G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, MD, 2nd ed., 1989).

[10] J.F. Grear, Operator coefficient methods for linear equations, Sandia Technical Report SAND89-8691, Sandia National Laboratories, 1989.

[11] M.H. Gutknecht, A completed theory of the unsymmetric Lanczos process and related algorithms, Part I, *SIAM J. Matrix Anal. Appl.* **13** (1992) 594–639.

[12] M.H. Gutknecht, A completed theory of the unsymmetric Lanczos process and related algorithms, Part II, IPS Research Report No. 90-16, ETH Zürich, 1990.

[13] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards* **49** (1952) 409–436.

[14] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Nat. Bur. Standards* **45** (1950) 255–282.

[15] L. Lempert, Recursion for orthogonal polynomials on complex domains, in: G. Alexits and P. Turán, Eds., *Fourier Analysis and Approximation Theory, Vol. II*, Colloq. Math. Soc. János Bolyai **19** (North-Holland, Amsterdam, 1978) 481–494.

[16] C.B. Moler and G.W. Stewart, An algorithm for generalized matrix eigenvalue problems, *SIAM J. Numer. Anal.* **10** (1973) 241–256.

[17] N.M. Nachtigal, L. Reichel and L.N. Trefethen, A hybrid GMRES algorithm for nonsymmetric linear systems, *SIAM J. Matrix Anal. Appl.*, to appear.

[18] B.N. Parlett, D.R. Taylor and Z.A. Liu, A look-ahead Lanczos algorithm for unsymmetric matrices, *Math. Comp.* **44** (1985) 105–124.

[19] Y. Saad, Variations of Arnoldi's method for computing eigenelements of large unsymmetric matrices, *Linear Algebra Appl.* **34** (1980) 269–295.

[?0] Y. Saad, Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems, *Math. Comp.* **42** (1984) 567–588.

[21] Y. Saad, Least squares polynomials in the complex plane and their use for solving nonsymmetric linear systems, *SIAM J. Numer. Anal.* **24** (1987) 155–169.

[22] Y. Saad and M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* **7** (1986) 856–869.

[23] P.E. Saylor and D.C. Smolarski, Computing the roots of complex orthogonal and kernel polynomials, *SIAM J. Sci. Statist. Comput.* **9** (1988) 1–13.

[24] D.C. Smolarski and P.E. Saylor, An optimum iterative method for solving any linear system with a square matrix, *BIT* **28** (1988) 163–178.

[25] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* **10** (1989) 36–52.

[26] E.L. Stiefel, Kernel polynomials in linear algebra and their numerical applications, *U.S. Nat. Bur. Standards Appl. Math. Ser.* **49** (1958) 1–22.

[27] G. Szegő, *Orthogonal Polynomials*, Amer. Math. Soc. Colloq. Publ. **23** (Amer. Mathematical Soc., Providence, RI, 4th ed., 1975).

[28] L.N. Trefethen, Pseudospectra of matrices, in: D.F. Griffiths and G.A. Watson, Eds., *Proc. 14th Dundee Biennial Conf. on Numerical Analysis*, to appear.

[29] J.H. Wilkinson, *The Algebraic Eigenvalue Problem* (Oxford Univ. Press, Oxford, 1965).