POLYNOMIAL APPROXIMATION TO THE INVERSE OF A LARGE MATRIX

MARK EMBREE[†], JOEL A. HENNINGSEN[‡], JORDAN JACKSON[†], AND RONALD B. MORGAN[‡]

Abstract. The inverse of a large matrix can often be accurately approximated by a polynomial of degree significantly lower than the order of the matrix. The iteration polynomial generated by a run of the GMRES algorithm is a good candidate, and its approximation to the inverse roughly follows the accuracy of the GMRES iteration. We investigate the quality of this approximation through theory and experiment, noting the practical need to add copies of some polynomial terms to improve stability. To mitigate storage and orthogonalization costs other approaches have appeal, such as polynomial preconditioned GMRES and deflation of problematic eigenvalues. Applications of such polynomial approximations include solving systems of linear equations with multiple right-hand sides (where the solutions to subsequent problems come simply by multiplying the polynomial against the new right-hand sides) and variance reduction in multilevel Monte Carlo methods.

Key words. linear equations, GMRES, polynomial preconditioning, matrix inverse, eigenvalues

AMS subject classifications. 65F10, 15A09

1. Introduction. Inverses of matrices occur throughout applications in mathematics, science, and engineering. One rarely needs to access the inverse explicitly; more typically one seeks the action of the inverse multiplied against a vector, i.e., the solution of a system of linear equations. For dense matrices this solution can be found via a matrix factorization; for large, sparse problems, one typically solves the system approximately with an iterative method that, effectively, multiplies an approximation of the inverse against a vector. Such algorithms are often motivated by invoking the Cayley–Hamilton theorem, which implies that the inverse of a matrix $A \in \mathbb{C}^{n\times n}$ can be expressed as a polynomial of degree $n-1$ (or less) in A. While it is impractical to compute this polynomial for large n, one can often find satisfactory approximations to A^{-1} using polynomials of significantly lower degree. We describe several ways to construct such an approximate polynomial representation of the inverse of a large matrix. We show that this polynomial can be accurate and useful, applying it to solve systems of linear equations with multiple right-hand sides. A second application, described in [22, 23], uses a polynomial approximation to an inverse to reduce variance in a Multilevel Monte Carlo sampling of the trace of the inverse. Given the ubiquity of the matrix inverse, we anticipate numerous other applications.

This work builds on the development of stable polynomial preconditioners for eigenvalue problems [13, 47] and linear systems [26] based on the GMRES algorithm [41]. Here, we focus on the quality of a polynomial $p(A)$ as an approximation of A^{-1} itself, addressing theoretical aspects but focusing on practical considerations required to make such approximations work. An implementation of the GMRES residual polynomial using its roots (harmonic Ritz values) is given in [13]. This approach is more stable than most previous methods for implementing the GMRES polynomial [1, 19, 25, 33] and cheaper to implement than another approach [47]. However, this polynomial can be prone to instabilities, prompting the stability control method in [13]: the polynomial is augmented with extra copies of roots near outstanding

[†]Department of Mathematics, Virginia Tech, Blacksburg, VA 24061 (embree@vt.edu, jgjacks4@vt.edu).

[‡]Department of Mathematics, Baylor University, Waco, TX 76798-7328 (Ronald_Morgan@baylor.edu, joel.henningsen32@gmail.com).

eigenvalues. This modification enables the practical use of higher degree polynomials.

When GMRES is applied to $Ax = b$, the residual vector can be written as

$$r = b - A\widehat{x} = b - Ap(A)b = \pi(A)b,$$

where $\widehat{x} = p(A)b$ is the approximate solution generated by GMRES, and $\pi(z) = 1 - zp(z)$ is the GMRES residual polynomial. For r to be small in norm, $p(A)$ should approximate A^{-1} (modulated by the vector b), and this p provides the starting point for our approximate inverse polynomial. In [26], an algorithm is given for multiplying $p(A)$ against a vector, given the roots of π. This implementation uses the technique from [13] for adding extra roots to π to promote stability, often making it possible to find a moderate degree polynomial p that achieves the goal of $p(A) \approx A^{-1}$.

We show that the accuracy of the polynomial $p(A)$ as an approximation to A^{-1} follows the GMRES residual. To reduce the cost of orthogonalization, we explore the use of a composite (or double) polynomial generated from polynomial preconditioned GMRES [26], along with the nonsymmetric Lanczos algorithm. For systems with multiple right-hand sides, once a polynomial has been found, the solution of additional systems simply requires multiplying $p(A)$ against the new right-hand sides. We also study a deflated version of the polynomial, which can give a lower degree polynomial.

Section 2 reviews some previous work that is needed for this project. Section 3 describes the use of GMRES to find a polynomial approximation to the inverse, and provides some theoretical results on the quality of this approximation and the location of the harmonic Ritz values. Section 4 investigates other approaches for constructing the polynomial: restarted GMRES, nonsymmetric Lanczos, and polynomial preconditioned GMRES. Section 5 uses the polynomial to solve systems with multiple right-hand sides, then Section 6 describes how to incorporate deflation. Finally, section 7 considers some examples that provide challenges for stabilization.

2. Review.

2.1. Polynomial preconditioning. For solving $Ax = b$, polynomial preconditioning is a way to transform the spectrum and thus improve convergence. With a polynomial p and right preconditioning, the linear equations problem becomes

$$Ap(A)y = b, \qquad x = p(A)y \tag{1}$$

Defining $\phi(z) \equiv zp(z) = \pi(z) - 1$, the preconditioned system of linear equations is

$$\phi(A)y = b.$$

We let dp denote the degree of p, so ϕ has degree $d\phi \equiv dp + 1$.

Much work has studied polynomial preconditioning; see, e.g., [1, 2, 3, 13, 15, 19, 21, 24, 25, 26, 37, 38, 39, 40, 44, 46, 47, 50, 51]. We highlight Thornquist's thesis [47], which constructs polynomial approximations to $(A - \mu B)^{-1}$ using GMRES (and several nonsymmetric Lanczos methods) to expedite shift-invert eigenvalue calculations. The use in [47] of non-optimal short-recurrence Krylov subspace methods to generate polynomial approximations of matrix inverses merits further investigation.

In [13, 26], starting with the GMRES residual polynomial π, the polynomial ϕ is chosen as $\phi(z) = 1 - \pi(z)$ and thus p is also determined. The roots of π are the harmonic Ritz values [28, 32, 34], and they are used to implement both polynomials ϕ and p. The paper [26] includes detailed algorithms that use the roots of π to apply $\phi(A)$ and $p(A)$ to vectors (respecting complex conjugate pairs of roots); see [26,

Algorithm 1 and Algorithm 3]. Thus the polynomials needed for polynomial preconditioning can be determined with one cycle of GMRES (frequently with a random starting vector [26]). Then GMRES can also be used to solve the linear equations as a part of polynomial preconditioned GMRES [26, Algorithm 4].

2.2. Stability for the polynomial. For a matrix with an eigenvalue that stands out from the rest of the spectrum, GMRES typically places a single harmonic Ritz value nearby, giving π a steep slope at that root that can lead to ill-conditioning and cause $p(A)$ evaluations to be unstable. To improve stability, extra copies of roots corresponding to outstanding eigenvalues can be added. Algorithm 1 (adapted from [13, p. A21]) shows a way to implement this procedure. For each root θ_k, one computes a diagnostic quantity called *pof*(k) (for "product of other factors") that measures the magnitude of $\pi(\theta_k)$ with the $(1 - z/\theta_k)$ term removed. When $\log_{10}(\mathit{pof}(k))$ exceeds some threshold *pofcutoff*, extra $(1-z/\theta_k)$ terms are appended to π. (In step 3, the value *pofcutoff* $= 4$ was used in [13]; in this paper, we use *pofcutoff* $= 8$.)

By construction, $p(z)$ interpolates $1/z$ at the roots of π. When we add a second copy of a root to stabilize $\pi(z)$, that root becomes a point where $p'(z)$ interpolates $(1/z)' = -1/z^2$. If θ is an m-fold root of π, one can show $p^{(j)}(\theta) = (-1)^j j!/\theta^{j+1} = \mathrm{d}^j/\mathrm{d}z^j\,(1/z)\big|_{z=\theta}$ for $j = 1, \ldots, m-1$: thus p interpolates $1/\theta$ and its first $m-1$ derivatives at $z = \theta$. Figure 1 shows how a second copy of a root can stabilize $\pi(z)$ and $p(z)$ in the proximity of an outlying eigenvalue.

2.3. Deflation. When solving linear systems, *deflation* refers to reducing the influence of small eigenvalues that tend to slow GMRES convergence. Deflation can be implemented by adding approximate eigenvectors to a subspace [29, 30], or by building a preconditioner from eigenvectors; see, e.g., [20, 35, 42]. Here we will deflate with Galerkin projection [23, 30, 31, 43, 45]; see Algorithm 2. This method can be applied before running a Krylov method, or between cycles of restarted GMRES. Section 6 provides details and examples.

3. Polynomial of A approximating A^{-1}.

3.1. Using GMRES to find an approximating polynomial. Let π denote a GMRES residual polynomial after sufficiently many iterations to solve a linear system,

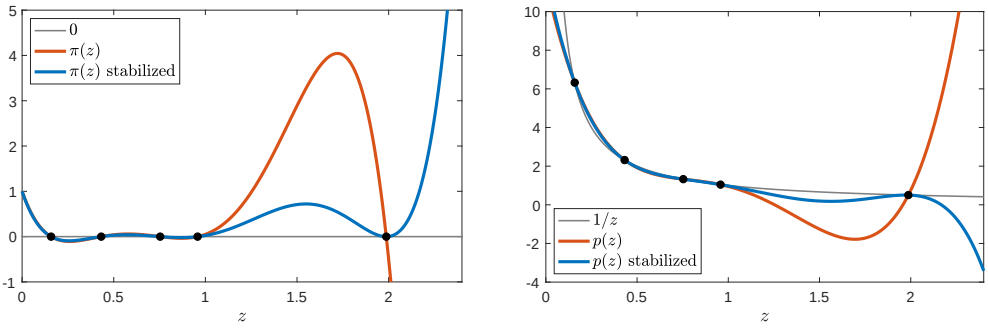

FIG. 1. *Adding extra copies of roots stabilize π and p. This A has real eigenvalues in $[0.1, 1]$ and one outlying eigenvalue at $\lambda = 2$. On the left, the degree $k = 5$ GMRES residual polynomial $\pi(z)$ (red) with roots at the black dots; note the steep slope at the root $\theta = 1.9869$ near the eigenvalue $\lambda = 2$. Adding an extra copy of that root leads to the degree $k+1 = 6$ polynomial (blue) that is small in a larger neighborhood of the root. On the right, the corresponding approximate inverse polynomial $p(z)$, which interpolates $1/z$ (gray line) at the black dots. The extra root has a similarly tonic effect on this polynomial, which now also interpolates the derivative of $1/\lambda$ at the extra root.*

Algorithm 1 Adding Roots to $\pi(\alpha)$ for Stability (from [13, p. A21])

1. Setup: Assume the $d\phi$ roots $(\theta_1, \ldots, \theta_{d\phi})$ of π have been computed and then sorted according to the modified Leja ordering [4, alg. 3.1]. (For high-degree polynomials and/or large magnitude roots, use sums of logs in place of products, to prevent overflow and underflow in the Leja ordering calculations.)

2. Compute *pof*(k)**:** For $k = 1, \ldots, d\phi$, compute $pof(k) = \prod_{i \neq k} |1 - \theta_k/\theta_i|$.

3. Add roots: Compute $\lceil \log_{10}(pof(k)) - pofcutoff)/14 \rceil$, for each k. Add that number of θ_k copies to the list of roots. Add the first to the end of the list; if there are others, space them evenly between the first and last occurrence of θ_k (keeping complex roots together).

Algorithm 2 Galerkin Projection

0. Let the current system of linear equations be $A(x - x_0) = r_0$.

1. Let V be an n-by-nev matrix with nev linearly independent columns.

2. Form $H = V^*AV$ and $c = V^*r_0$.

3. Solve $Hd = c$ and let $\widehat{x} = Vd$.

4. The new approximate solution is $x_e = x_0 + \widehat{x}$, with residual $r = r_0 - A\widehat{x} = r_0 - AVd$.

and define p via $\pi(z) = 1 - zp(z)$. Examples will show that it is possible for $p(A)$ to be a good approximation to A^{-1}, and that the accuracy of this approximation typically tracks the GMRES residual norm. A theorem backs this up for the case of normal A.

In developing $p(A)$, we presume the GMRES starting vector is not (nearly) missing components in any eigenvectors. Missing components, particularly at extreme eigenvalues, could cause the polynomial to be inaccurate there, and thus ineffective at approximating the inverse. When solving for multiple right-hand sides (Sections 5 and 6), if the first right-hand side has missing components, then a random vector should be used to develop $p(A)$.

When a standard preconditioner is available for A, we can still get an approximation to A^{-1}, but it will not be a polynomial of A alone. Looking at the system with standard right preconditioning, where M is an approximation to A,

$$AM^{-1}w = b, \qquad \widehat{x} = M^{-1}w,$$

we see $r = b - A\widehat{x} = b - AM^{-1}p(AM^{-1})b = \big(I - AM^{-1}p(AM^{-1})\big)b$, where p comes from GMRES applied to the standard preconditioned system. For $\|r\|$ to be a small, A^{-1} should be approximated by $M^{-1}p(AM^{-1})$. Thus standard preconditioning can be incorporated into this framework, though we do not pursue this further here.

Our first example shows how polynomial approximation of the inverse can work with a matrix from a simple application problem.

Example 1. Consider the standard second-order finite difference discretization of the convection-diffusion equation $-u_{xx} - u_{yy} + 2u_x = f$ on a uniform grid over the unit square $[0, 1] \times [0, 1]$, with homogeneous Dirichlet boundary conditions. Grid size $1/50$ gives A of order $n = 2500$. GMRES is run with a random starting vector (normal entries, scaled to unit norm). Figure 2 compares GMRES convergence to the relative accuracy of the polynomial, $\|A^{-1} - p(A)\|/\|A^{-1}\|$. Notice that $p(A)$ can accurately approximate A^{-1}; the relative accuracy of $p(A)$ keeps pace with the GMRES residual norm, typically about an order of magnitude behind. The GMRES residual norm goes below 10^{-12} at $k = 217$, where $p(A)$ has relative accuracy of 5.1×10^{-12}.

Figure 3 compares the polynomial $p(z)$ to $f(z) = 1/z$. For $p(A)$ to approximate

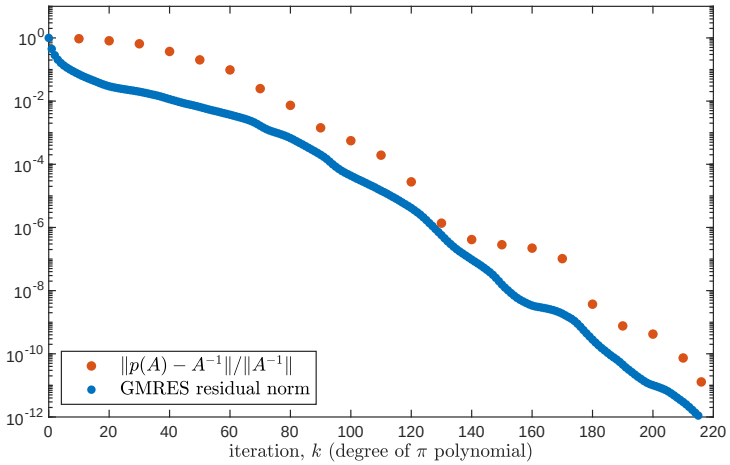

FIG. 2. *Example 1: The matrix is size $n = 2500$ from the convection-diffusion equation $-u_{xx}-u_{yy}+2u_x = f$. The relative accuracy, $\|A^{-1}-p(A)\|/\|A^{-1}\|$, of the polynomial approximation to the inverse is shown every 10 GMRES iterations, and is compared to the GMRES residual norm.*

A^{-1} well, $p(z)$ must be close to $1/z$ at every eigenvalue of A (made precise later in this section). At iteration $k = 10$, p looks to be a good approximation over the entire spectrum of A (top plot). However, near the origin (bottom plot) we see that the $k = 10$ polynomial does not approximate $1/z$ well at the small eigenvalues. At iteration $k = 50$ it is better, but still off a bit. The polynomial at $k = 175$ (computed at the point where the GMRES residual hits 10^{-9}) is accurate at all eigenvalues.

In Figure 4 we adjust parameters in the differential equation to make A more nonsymmetric, then also indefinite. For $-u_{xx}-u_{yy}+\alpha u_x+\beta u_y-\gamma^2 u = f$, we first let $\alpha = 25$, $\beta = 10$ and $\gamma = 0$. The accuracy with which $p(A)$ approximates A^{-1} (upper-

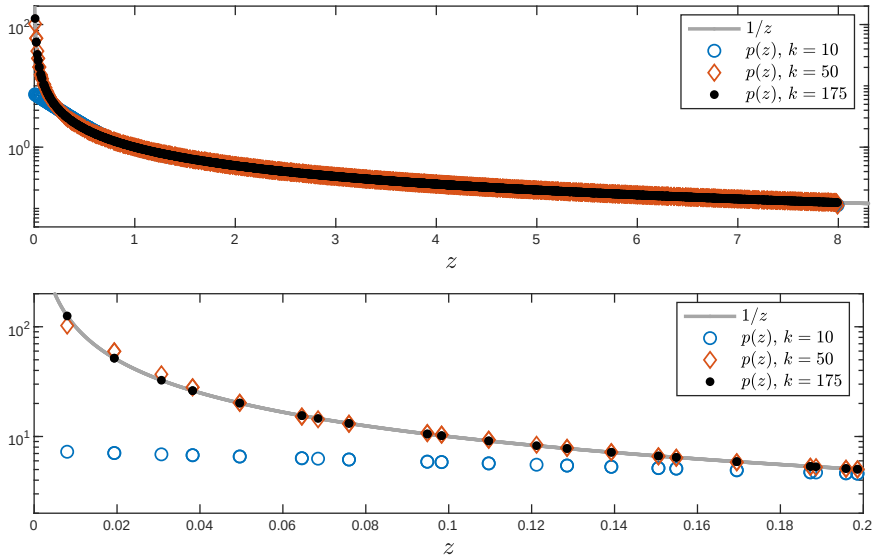

FIG. 3. *Convection-diffusion example from Figure 2 (dimension $n = 2500$), showing how the polynomial $p(z)$ approximates $1/z$ at the eigenvalues of A: the gray line shows $1/z$; the markers show $p(\lambda)$ for the eigenvalues λ of A, for three different degree polynomials p of degree $k-1$.*

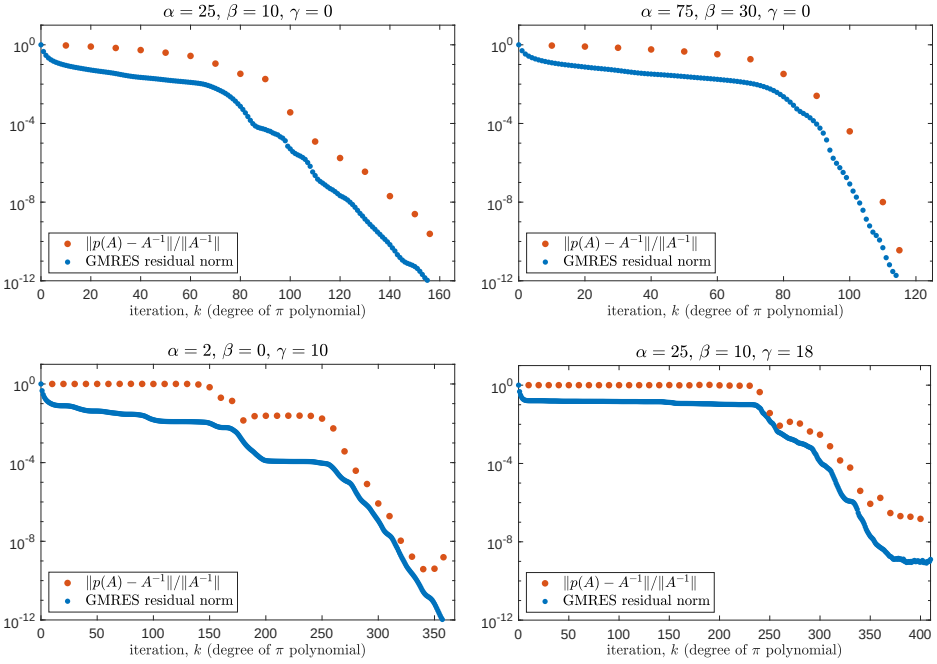

FIG. 4. *Matrix of size $n = 2500$ from the convection-diffusion equation $-u_{xx} - u_{yy} + \alpha u_x + \beta u_y - \gamma^2 u = f$. The relative accuracy of $p(A)$ is compared to the GMRES residual for varying degrees of nonnormality and indefiniteness. (The last few red dots in the bottom plots likely reflect some numerical instabilities.)*

left) follows the GMRES residual norm, though not quite as closely as in Figure 2 with a nearly symmetric A. The polynomial remains effective with more nonnormality (upper-right) and with indefiniteness (lower-left). However, the accuracy is more erratic when the matrix is both significantly nonnormal and indefinite (lower-right).

3.2. Polynomial approximation accuracy and the GMRES residual.

THEOREM 1. *Suppose $A \in \mathbb{C}^{n\times n}$ is invertible and $Ax = b$ for some unit vector $b \in \mathbb{C}^n$. Let $\widehat{x} = p(A)b$ be an approximation to x for some polynomial p, and let $\kappa(A) = \|A\|\|A^{-1}\|$. Then*

$$\frac{\|r\|}{\kappa(A)} \;\le\; \frac{\|x-\widehat{x}\|}{\|A^{-1}\|} \;\le\; \frac{\|A^{-1}-p(A)\|}{\|A^{-1}\|}. \tag{2}$$

Proof. Since $r = b - A\widehat{x} = b - Ap(A)b = A(x-\widehat{x})$, we can write

$$x - \widehat{x} \;=\; A^{-1}r \;=\; \big(A^{-1} - p(A)\big)b.$$

Since $\|b\| = 1$, we have $\|x-\widehat{x}\| \le \|A^{-1} - p(A)\|$. Divide by $\|A^{-1}\|$ to get the second inequality in (2). The first follows from $\|r\| = \|A(x-\widehat{x})\| \le \|A\|\|x-\widehat{x}\|$. ∎

This result shows that, for well-conditioned problems, the norm of the GMRES residual can serve as an indicator for the relative error in the polynomial approximation of the inverse: If the norm of the residual remains large, we cannot expect the corresponding polynomial to approximate the inverse with any accuracy.

Next, for normal coefficient matrices, we can also use the residual to get an *upper bound* on the error of the polynomial approximation. (Recall that A is normal if $A^*A = AA^*$ or, equivalently, if A has an orthonormal basis of eigenvectors.)

THEOREM 2. *Suppose $A \in \mathbb{C}^{n\times n}$ is an invertible normal matrix and $Ax = b$ for some unit vector $b \in \mathbb{C}^n$. Let $\widehat{x} = p(A)b$ for some polynomial p, and let $A = Z\Lambda Z^*$ be a unitary diagonalization of A with orthonormal eigenvectors $z_1, \ldots, z_n$ and corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$. Expand $b = Z(Z^*b) = \sum \beta_i z_i$. Then*

$$\frac{\|A^{-1} - p(A)\|}{\|A^{-1}\|} \leq \frac{\|r\|}{\min|\beta_i|}. \tag{3}$$

Proof. Again writing $r = b - A\widehat{x} = b - Ap(A)b$, we have

$$\begin{aligned}\|A^{-1}\|\|r\| &\geq \|A^{-1}r\| = \|(A^{-1} - p(A))b\| \\ &= \|Z(\Lambda^{-1} - p(\Lambda))Z^*b\| \;=\; \|(\Lambda^{-1} - p(\Lambda))[\beta_1 \ldots \beta_n]^T\|,\end{aligned}$$

due to the unitary invariance of the 2-norm. We can then bound

$$\begin{aligned}\|A^{-1}r\|^2 &= \sum_{i=1}^{n} \left|\Big(\frac{1}{\lambda_i} - p(\lambda_i)\Big)\beta_i\right|^2 \\ &\geq \max\left|\frac{1}{\lambda_i} - p(\lambda_i)\right|^2 \cdot \min|\beta_i|^2 \;=\; \|A^{-1} - p(A)\|^2 \cdot \min|\beta_i|^2,\end{aligned}$$

since the norm of a function of a normal matrix is the maximum magnitude of the function on the eigenvalues of that matrix. Rearranging the last expression gives

$$\frac{\|r\|}{\min|\beta_i|} \geq \frac{\|A^{-1} - p(A)\|}{\|A^{-1}\|}.$$

∎

By the bound (3), for normal A a small GMRES residual norm implies a small relative error in the A^{-1} approximation, *provided b is not deficient in any eigenvectors.* To make such deficiencies unlikely, one can take b to have normally distributed random entries, so the entries of $\beta = Z^*b$ will also be normally distributed for unitary Z.

3.3. Bounding $\|A^{-1} - p(A)\|$. The quality with which $p(A)$ approximates A^{-1} depends on how well $p(\lambda)$ approximates $1/\lambda$ on the spectrum of A, denoted by $\sigma(A)$, or some larger set depending on the departure of A from normality. Consider the *numerical range* of A, which is the set of all Rayleigh quotients [17, chap. 1],

$$W(A) = \left\{\frac{v^*Av}{v^*v} : 0 \neq v \in \mathbb{C}^n\right\},$$

and the ε-pseudospectrum of A [48],

$$\sigma_\varepsilon(A) = \left\{z \in \mathbb{C} : \|(zI - A)^{-1}\| > 1/\varepsilon\right\}.$$

The following proposition follows from conventional bounds on functions of matrices, and all have analogs for bounding the residual polynomial in GMRES [12].

PROPOSITION 3. *Suppose $A \in \mathbb{C}^{n\times n}$ is invertible, and let p be any polynomial.*

1. *If A is diagonalizable, $A = Z\Lambda Z^{-1}$, then with $\kappa(Z) = \|Z\|\|Z^{-1}\|$,*

$$\|A^{-1} - p(A)\| \leq \kappa(Z) \max_{\lambda\in\sigma(A)} \left|\frac{1}{\lambda} - p(\lambda)\right|.$$

2. *If $0 \notin W(A)$, then, by the Crouzeix–Palencia theorem [10],*

$$\|A^{-1} - p(A)\| \leq (1 + \sqrt{2}) \max_{\lambda\in W(A)} \left|\frac{1}{\lambda} - p(\lambda)\right|.$$

3. *If $\varepsilon > 0$ is sufficiently small that $0 \notin \sigma_\varepsilon(A)$, then*

$$\|A^{-1} - p(A)\| \le \frac{L_\varepsilon}{2\pi\varepsilon} \max_{\lambda \in \sigma_\varepsilon(A)} \Big|\frac{1}{\lambda} - p(\lambda)\Big|,$$

where L_ε denotes the length of the boundary of $\sigma_\varepsilon(A)$.

3.4. Harmonic Ritz values. The roots of $\pi(z) = 1 - zp(z)$ are precisely the points where $p(z)$ interpolates $1/z$. Thus in light of Proposition 3, to understand how well $p(A)$ approximates A^{-1} one naturally asks, *Where are these roots of π located?* Thus we briefly discuss properties of the harmonic Ritz values and their location.

At step k, GMRES computes the approximation $\widehat{x}$ from the Krylov subspace

$$\mathcal{K}_k(A,b) \;=\; \operatorname{span}\{b, Ab, \ldots, A^{k-1}b\} \;=\; \{\psi(A)b : \deg(\psi) < k\}.$$

Conventional implementations of GMRES progressively build an orthonormal basis $\{v_1, \ldots, v_k\}$ for $\mathcal{K}_k(A,b)$. This process can be compactly summarized in the expression

$$AV_k = V_{k+1}H_{k+1,k} = V_kH_{k,k} + h_{k+1,k}v_{k+1}e_k^*, \tag{4}$$

where the $(k+1)\times k$ upper Hessenberg matrix $H_{k+1,k}$ (and its top $k\times k$ block, $H_{k,k}$) collect the coefficients from the orthogonalization process. Premultiplying (4) by V_k^* gives $H_{k,k} = V_k^*AV_k$. (The eigenvalues of $H_{k,k} \in \mathbb{C}^{k\times k}$ are called *Ritz values*; they are Rayleigh–Ritz eigenvalue estimates for A from the Krylov subspace $\mathcal{K}_k(A,b)$.)

The GMRES approximation $\widehat{x} \in \mathcal{K}_k(A,b)$ is selected to minimize the 2-norm of the residual, $\|r\| = \|b - A\widehat{x}\|$, and so basic least squares theory implies that the residual $r = b - A\widehat{x}$ is orthogonal to the space $A\mathcal{K}_k(A,b)$ from which b is approximated.

We write $r = \pi(A)b$, and seek to characterize the roots of π. The following derivation comes from [8, 27]. Suppose π has a root θ, which cannot be zero since $\pi(0) = 1$. Factor out this root: $\pi(z) = (1 - z/\theta)\widetilde{\pi}(z)$ for some $\widetilde{\pi}$ with $\deg(\widetilde{\pi}) < k$. Since r is orthogonal to $A\mathcal{K}_k(A,b) = \operatorname{range}(AV_k)$ and $\widetilde{\pi}(A)b \in \mathcal{K}_k(A,b)$, we can write

$$\begin{aligned} 0 \;=\; (AV_k)^*r \;&=\; V_k^*A^*\big(I - (1/\theta)A\big)\widetilde{\pi}(A)b \\ &=\; V_k^*A^*\big(I - (1/\theta)A\big)V_kc \end{aligned} \tag{5}$$

for some $c \in \mathbb{C}^k$. Rearrange this last equation to get $V_k^*A^*AV_kc = \theta\, V_k^*A^*V_kc$, a generalized eigenvalue problem that characterizes the roots of the GMRES residual polynomial π. Using the Arnoldi identity (4), we can obtain the more compact form

$$H_{k+1,k}^*H_{k+1,k}\, c = \theta H_{k,k}^*c. \tag{6}$$

Since the coefficient matrices $H_{k+1,k}^*H_{k+1,k}$ and $H_{k,k}^*$ are $k\times k$, this generalized eigenvalue problem can have up to k finite eigenvalues, the roots of π. When $H_{k,k}$ is singular the generalized eigenvalue problem can have fewer than k finite eigenvalues; this case corresponds to the stagnation of GMRES at step k. When $H_{k,k}$ is invertible, (6) can be reduced to the standard eigenvalue problem

$$\Big(H_{k,k} + |h_{k+1,k}|^2 f_k e_k^*\Big)c = \theta c, \qquad f_k := H_{k,k}^{-*}e_k. \tag{7}$$

The roots of π, characterized by (6), are called *harmonic Ritz values* [16, 28, 32, 34], since $1/\theta$ *is a Rayleigh–Ritz eigenvalue estimate for A^{-1} from the subspace* $A\mathcal{K}_k(A,b) = \operatorname{range}(AV_k)$. To see this, note that (5) is equivalent to

$$(AV_k)^*A^{-1}(AV_k)c = \frac{1}{\theta}(AV_k)^*(AV_k)c,$$

a generalized Rayleigh quotient for A^{-1}: Unlike standard Ritz values (the eigenvalues of $H_{k,k}$), *harmonic Ritz values are not shift invariant*: their location depends on the proximity of the spectrum of A to the origin. Since the reciprocals of the harmonic Ritz values $\{\theta_j\}_{j=1}^k$ are Ritz values for A^{-1}, we have $1/\theta_j \in W(A^{-1})$, that is,

$$\theta_j \in \{1/z : z \in W(A^{-1})\} = \left\{ \frac{v^*v}{v^*A^{-1}v} : 0 \neq v \in \mathbb{C}^n \right\}.$$

Look ahead to Figure 6 to see an example of $1/W(A^{-1})$. This set contains $z = \infty$ when $0 \in W(A^{-1})$, which implies $0 \in W(A)$ and thus the ability of the first step of GMRES to stagnate. (The potential for stagnation at later iterations can be described via higher-dimensional generalizations of the numerical range [14, Theorem 2.7].)

The case of Hermitian A already shows the subtle nature of harmonic Ritz values. If A is positive definite with eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, the harmonic Ritz values obey Cauchy interlacing (see, e.g., [36]): if $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_k$, then

$$\lambda_j \leq \theta_j \leq \lambda_{n-j+k}, \qquad j = 1, \ldots, k.$$

In contrast, if A is indefinite (and thus $0 \in W(A)$) with eigenvalues

$$\lambda_{-m} \leq \cdots \leq \lambda_{-1} < 0 < \lambda_1 \leq \cdots \leq \lambda_p,$$

one can say

$$\theta_j \in (-\infty, \lambda_{-1}] \cup [\lambda_1, \infty), \qquad j = 1, \ldots, k.$$

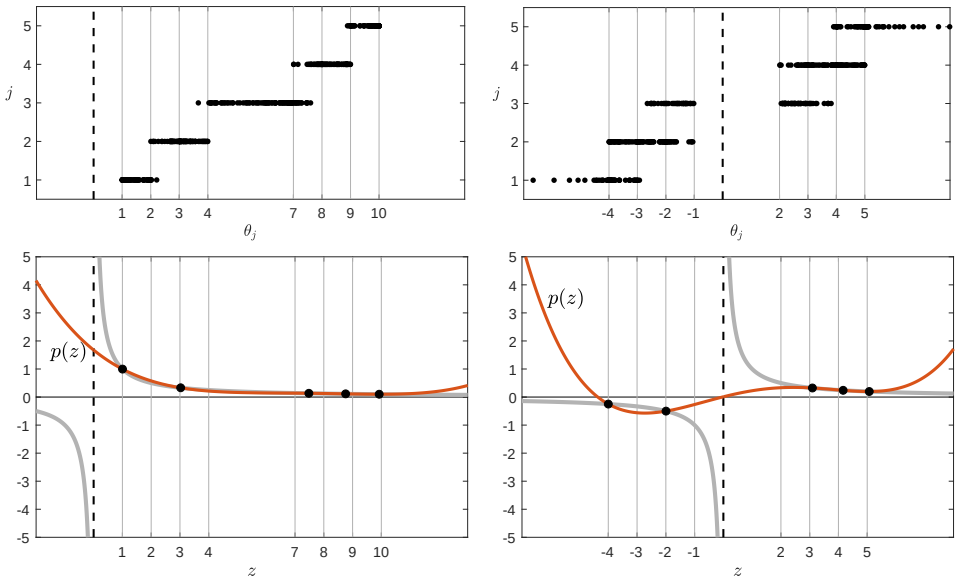

FIG. 5. *The top plots show the harmonic Ritz values from 100 trials involving random Krylov subspaces of dimension $k = 5$ for Hermitian matrices of dimension $n = 8$, sorted $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_5$. Eigenvalues are marked by gray vertical lines; the origin is denoted by the vertical black dashed line. The positive definite matrix on the left has eigenvalues $\{1, 2, 3, 4, 7, 8, 9, 10\}$, and the harmonic Ritz values obey Cauchy interlacing. The indefinite problem on the right has the same eigenvalues, only shifted left: $\{-4, -3, -2, -1, 2, 3, 4, 5\}$. Notice the absence of harmonic Ritz values in the interval $(-1, 2)$; numerous θ_j values are beyond the axis limits for this case. The two plots would be identical (up to the shift) if they showed standard Ritz values, which are shift invariant. The bottom plots show how $p(z)$ (red line) interpolates $1/z$ (gray line) for one of the trials. Notice that approximating $1/z$ at eigenvalues on both sides of the origin (indefinite problem on the right) is much more challenging.*

Thus *no* harmonic Ritz values call fall in the interval $(\lambda_{-1}, \lambda_1)$ containing the origin. Figure 5 gives a simple illustration contrasting the positive definite and indefinite cases. For a more detailed discussion of the Hermitian case, see [6].

The following result bounds the magnitudes of harmonic Ritz values via geometric means of the trailing singular values of A. It follows by applying an upper bound for Ritz values [9, Theorem 2.3] to bound $1/\theta_j$ as a Ritz value of A^{-1}.

THEOREM 4. *Suppose $A \in \mathbb{C}^{n\times n}$ and let $\{\theta_j\}_{j=1}^k$ denote the roots of the degree-k GMRES residual polynomial, ordered by increasing magnitude: $|\theta_1| \le |\theta_2| \le \cdots \le |\theta_k|$ (allowing for the possibility that some $\theta_j = \infty$). Then*

$$|\theta_j| \ge \left(s_n \cdots s_{n-j+1}\right)^{1/j}, \qquad j = 1, \ldots, k,$$

where $s_1 \ge s_2 \ge \cdots \ge s_n$ denote the singular values of A.

The $k = 1$ case of this theorem ensures that no harmonic Ritz values can fall strictly inside the disk of radius $s_n = 1/\|A^{-1}\|$ centered at the origin. For larger k the bounds quantify the notion that *there cannot be more small magnitude harmonic Ritz values than the number of small singular values of A*. The development of finer containment regions for harmonic Ritz values is an interesting but difficult problem. When coupled with Proposition 3, such results could illuminate how well $p(A)$ approximates A^{-1}.

Figure 6 contrasts standard Ritz values (eigenvalues of $H_{k,k}$) with harmonic Ritz values (eigenvalues of $H_{k,k} + |h_{k+1,k}|^2 f_k e_k^*$) when A is the circulant shift matrix of order $n = 7$ (ones on the superdiagonal and bottom-left entry; zeros elsewhere), a unitary matrix that is notoriously difficult for GMRES [7]. The Ritz values all fall in $W(A)$, in this case the convex hull of the spectrum, located *inside the unit disk*. In contrast, the harmonic Ritz values fall in $1/W(A^{-1})$, a nonconvex set *exterior to the unit disk* (except at the eigenvalues).

3.5. Further tests.

Example 2. Returning to the convection–diffusion example, we explore how the accuracy of the polynomial approximation of A^{-1} obtained from GMRES depends on b. For a baseline we use the Laplacian $-u_{xx} - u_{yy} = f$, giving a symmetric A. The left plot in Figure 7 shows the results of 100 trials with different random b. The

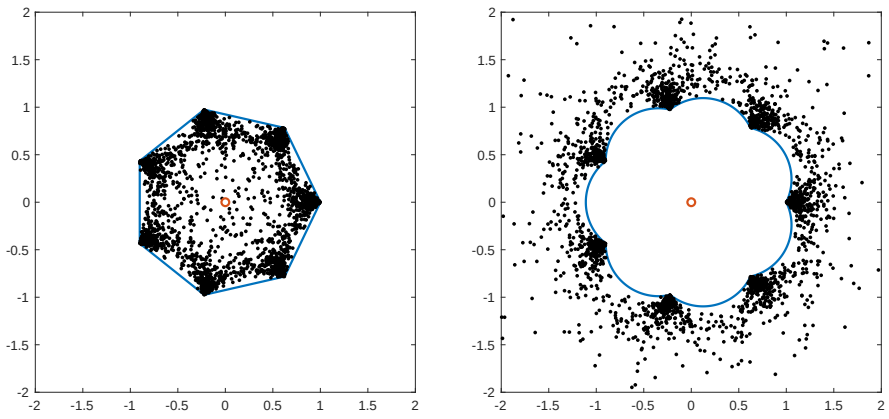

FIG. 6. *Ritz values (left) and harmonic Ritz values (right) from 5-dimensional random complex Krylov subspaces of the circulant shift ($n = 7$), 500 trials. (83 of the 2500 harmonic Ritz values fall beyond the axis.) The origin is marked in red; the blue lines show the boundaries of $W(A)$ (left – an exterior bound on Ritz values) and $1/W(A^{-1})$ (right – an interior bound on harmonic Ritz values).*

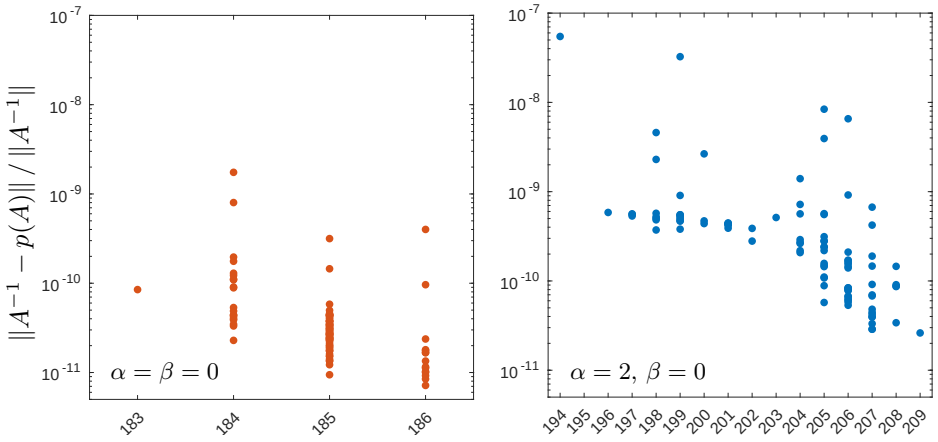

FIG. 7. *Matrix is of size $n = 2500$ from the convection–diffusion equation $-u_{xx} - u_{yy} + \alpha u_x + \beta u_y = f$ with $\alpha = \beta = 0$ (left) and $\alpha = 2$, $\beta = 0$ (right). Here $p(A)$ is determined by GMRES, run until the relative residual norm is below 10^{-11}. The relative accuracy $\|A^{-1} - p(A)\|/\|A^{-1}\|$ and the number of GMRES steps (number of harmonic Ritz values) are shown for 100 random b.*

polynomial is determined by running GMRES to relative residual norm of 10^{-11}. The plot shows both the degree of the final GMRES residual polynomial $\pi(z) = 1 - zp(z)$ and the relative accuracy of $p(A)$ as an approximation of A^{-1}. The degree of π is quite consistent, as is the accuracy of $p(A)$. The b vector that gives the worst result has a small component (3.3×10^{-4}) in the eigenvector corresponding to the eleventh eigenvalue ($\lambda = 0.06811$). The GMRES residual polynomial π is not very accurate at that eigenvalue; it differs from the needed value of 0 by 1.6×10^{-8}. As Theorem 2 suggests, if b has a small component in an eigenvector, then $p(z)$ need not be accurate at that eigenvalue despite the small residual norm.

The right plot in Figure 7 use the same convection-diffusion equation $-u_{xx} - u_{yy} + 2u_x = f$ that began Example 1, giving A with a mild departure from normality. Again we run GMRES to tolerance 10^{-11} using the same 100 random b used for the symmetric A. Now the accuracy of the A^{-1} approximation is more varied, though most cases are better than 10^{-9}. The poorest two results (relative accuracy worse than 10^{-8}) both have the polynomial less accurate at one of two nearly equal eigenvalues (the second and third eigenvalues, 0.0193339 and 0.0193361). For the worst case, π is not accurate at the larger of the two nearly multiple eigenvalues: π needs to be nearly 0 at all eigenvalues, but it differs from 0 by 1.3×10^{-7} at that eigenvalue.

Example 3. We examine the effect of ill-conditioning on the accuracy of the polynomial approximation of the inverse. To focus on this effect, we take A to be a positive definite diagonal matrix with no outstanding eigenvalues. (No extra stabilization roots are required in this experiment.) For dimension $n = 2501$ we set

$$\widehat{A} = \mathrm{diag}(1^p, 2^p, \ldots, 1251^p, 2 \cdot (1251^p) - 1250^p, 2 \cdot (1251^p) - 1249^p, \ldots, 2 \cdot (1251^p) - 1^p)$$

and scale $A = \widehat{A}/(1250.5)^{p-1}$. As the parameter $p \geq 0$ increases, the conditioning grows and the eigenvalues evolve from clustered in the middle of the spectrum ($p < 1$) to clustered at both ends ($p > 1$). We take 20 values of $p \in [0.55, 1.8]$, in each case running GMRES with the same b (a random normal vector, scaled to unit norm).

This example shows how the degree of the polynomial varies with the condition number, and that a high degree polynomial can be effective. Figure 8 shows the

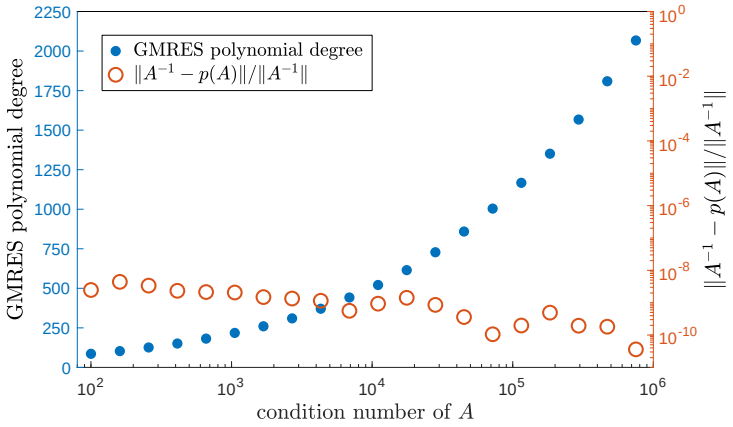

FIG. 8. *Example 3: diagonal matrices* ($n = 2501$) *with no outstanding eigenvalues. The degree of the GMRES polynomial when the residual goes below* 10^{-10} *and the corresponding relative error in the polynomial approximation of* A^{-1} *is plotted versus the condition number.*

degree needed for full GMRES to converge to residual norm below 10^{-10}. The relative accuracy of $p(A)$ ranges from a maximum of 4.4×10^{-9} for $p \approx 0.616$ to a minimum of 3.6×10^{-11} for $p = 1.8$. Even with ill-conditioned problems, $p(A)$ can accurately approximate A^{-1}, though the degree of p grows with the ill-conditioning.

4. Alternative Methods for Finding the Polynomial. Finding the polynomial p that approximates A^{-1} can be expensive due to the need to run full (non-restarted) GMRES, possibly for many iterations. The orthogonalization cost and storage increase as the iteration proceeds. We discuss three ways of reducing this cost: generating the polynomial p from restarted GMRES, a double polynomial formed by polynomial preconditioned GMRES, and the nonsymmetric Lanczos process.

4.1. Restarted GMRES. The restarted GMRES algorithm [41] can significantly reduce the orthogonalization cost, restricting the GMRES optimization to Krylov subspaces of fixed dimension $m \ll n$. However, restarted GMRES often takes more iterations than full GMRES, so a higher degree polynomial is needed.

When GMRES(m) is run for c cycles, the overall residual polynomial is a product of the residual polynomials for each cycle, $\pi(z) = \pi_1(z) \cdots \pi_c(z)$, where each π_j is a degree-m polynomial. The roots of $\pi(z)$ are all of the roots of the residual polynomials (harmonic Ritz values) from each GMRES cycle put together. The polynomial p is defined via $\pi(z) = 1 - zp(z)$. Then $p(A)$ can be multiplied against a vector using [26, Algorithm 3] with the list of roots from the overall π. When each restart occurs at the end of a cycle of m iterations, m of these roots become "locked in", and will henceforth be roots of the π that is built up over future cycles. (In contrast, for full GMRES *all* roots of π can change at each iteration.) The harmonic Ritz values may nearly recur in cyclic patterns across restarted GMRES cycles [5, 52], which could limit the effectiveness of this approach to designing polynomial approximations to the inverse. (In the "hybrid GMRES" approach [33], m iterations of GMRES are run, and the polynomial is constructed from $\pi(z) = (1 - z/\theta_1)^c \cdots (1 - z/\theta_m)^c$. The resulting approximation $p(z)$ will interpolate $1/z$ and its first $c - 1$ derivatives at $\theta_1, \ldots, \theta_m$.)

4.2. Double polynomials. An alternative approach to building high-degree polynomials while controlling orthogonalization costs uses the composition of two polynomials, as generated by polynomial preconditioned GMRES. We call such an

approximation to the inverse a *double polynomial* [13, 26]. The double polynomial often needs to be of higher degree than a single GMRES polynomial to deliver the same accuracy, but it can be of lower degree than the restarted GMRES polynomial. In general, the double polynomial makes approximation of A^{-1} much more practical.

We describe how to build the double polynomial from polynomial preconditioned GMRES (Subsection 2.1). First, select the degree of the ϕ polynomial in polynomial preconditioned GMRES. We regard this as an inner polynomial, so we call it ϕ_{in} and its degree $d\phi_{\text{in}}$. Run GMRES on A for $d\phi_{\text{in}}$ iterations, and compute the roots of the GMRES residual polynomial π_{in}. These roots define the polynomials ϕ_{in} and p_{in} according to $\pi_{\text{in}}(z) = 1 - \phi_{\text{in}}(z) = 1 - zp_{\text{in}}(z)$. Next, run GMRES a second time ("outer GMRES"), now on the polynomial preconditioned system $Ap_{\text{in}}(A)y = b$ (or $\phi_{\text{in}}(A)y = b$), solving to the required tolerance. (The approximate solution of the linear system is $x = p_{\text{in}}(A)y$.) Compute the roots of the residual polynomial from this outer GMRES run; they define the polynomials p_{out} and ϕ_{out}. The degree of ϕ_{out}, say $d\phi_{\text{out}}$, equals the number of outer GMRES iterations. The overall polynomial p that approximates the inverse is $p(z) = p_{\text{in}}(z)p_{\text{out}}(\phi_{\text{in}}(z))$, of degree $d\phi_{in} \times d\phi_{out} - 1$.

4.3. Nonsymmetric Lanczos. The nonsymmetric Lanczos algorithm can solve linear equations in various ways, the most straightforward of which is the BiConjugate Gradient (BiCG) method [40]. The BiCG residual polynomial π has roots at the eigenvalues of the tridiagonal matrix built by the Lanczos process, which determine the inverse approximating polynomial p through $\pi(z) = 1 - zp(z)$. (Future work should explore more sophisticated approaches based on nonsymmetric Lanczos, building on Thornquist's investigation of BiCGSTAB and TFQMR [47].)

4.4. Comparison. *Example 4.* We return to the matrix from Example 1 of dimension 2500 corresponding to $-u_{xx} - u_{yy} + 2u_x = f$. Figure 9 shows results for polynomials found by GMRES, restarted GMRES, the double polynomial from polynomial preconditioned GMRES, and nonsymmetric Lanczos. The linear equation residual norms are shown, along with the relative accuracy $\|A^{-1} - p(A)\|/\|A^{-1}\|$.

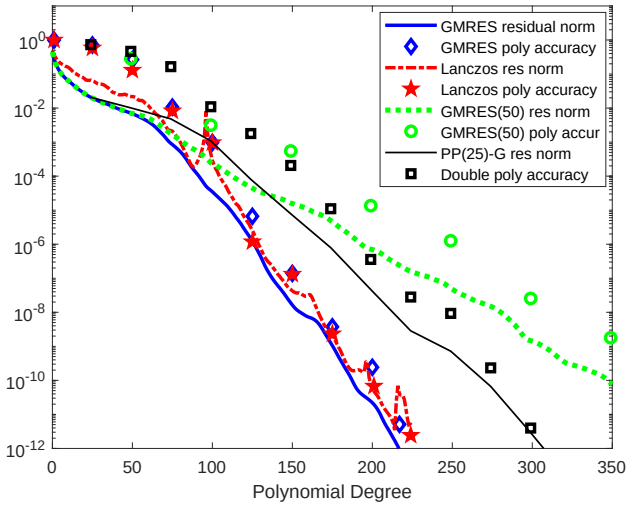

FIG. 9. *Comparison of four methods for building polynomial approximations to A^{-1}, for the convection–diffusion problem from Example 1 ($n = 2500$). The lines show the residual norms of the linear system; the markers show the relative accuracy of the polynomial approximation to A^{-1}.*

Nonsymmetric Lanczos has low orthogonalization costs and finds polynomials that are almost as low degree for a given accuracy as those from full GMRES. However, it requires more matrix-vector products, multiplying with both A and A^* at each iteration. The method can also be unstable, and for an indefinite matrix, Ritz values can fall near the origin. The accuracy of the double polynomial generally stays ahead of the restarted GMRES polynomial, and requires even less orthogonalization. Thus we will mostly use the double polynomial in the rest of this paper.

5. Linear Equations with Multiple Right-hand Sides. Given a good polynomial approximation to A^{-1}, one can quickly solve additional linear systems: simply multiply each right-hand side by $p(A)$, as in Algorithm 3. One simply needs to represent p via roots of π, whether obtained from full GMRES, or one of the alternatives discussed in the last section. For double polynomial preconditioning the process is a bit more subtle, and so is detailed in Algorithm 4.

Example 5. Consider the discretization of $-u_{xx} - u_{yy} + 2u_x - 10^2 u = f$ on the unit square with 200 interior grid points in each direction, giving A of dimension $n = 40{,}000$. The $-10^2 u$ term makes the matrix indefinite and poses a challenge for conventional solvers like restarted GMRES and BiCGstab [49]. We seek to solve 10 systems with random normal right-hand sides (scaled to unit norm) to residual norm 10^{-8}. The first two rows of Table 1 show the results of standard methods. Restarted GMRES(100) with a limit of 40,000 cycles not only takes over 28 hours; it only solves five systems to accuracy below 10^{-8} (others as high as 6×10^{-3}). BiCGStab with a limit of 100,000 iterations takes 380 seconds and only solves four systems to 10^{-8} (others as high as 3.5×10^{-4}). The remaining rows show results for polynomial inverse approximation. First, running full GMRES to relative residual tolerance of 10^{-11} creates a degree 1345 polynomial p (which includes two extra roots added for stability control; see Section 7). The next nine systems $Ax^j = b^j$ are solved by multiplying $p(A)b^j$, and all reach residual norms below 6×10^{-9}. Finding p and solving the

Algorithm 3 Solve multiple right-hand sides with a polynomial from GMRES

1. **First right-hand side system.** Solve the first system to the requested relative residual tolerance with full GMRES. Find the harmonic Ritz values from the last GMRES iteration, and use them to build a polynomial approximation $p(A)$ to A^{-1}. Add roots for stability, if necessary (Algorithm 1).
2. **Other right-hand side systems.** Apply $p(A)$ to the right-hand sides of the other systems. For the systems $Ax^j = b^j$, $j = 2 : nrhs$, compute $x^j = p(A)b^j$ using [26, Algorithm 3].

Algorithm 4 Solve multiple right-hand sides with a double polynomial

1. **First right-hand side.** Choose the degree $d\phi_{\text{in}}$ of the inner polynomial ϕ_{in}. Run $d\phi_{\text{in}}$ iterations of GMRES on the first right-hand side and then compute the roots of the residual polynomial π_{in}. These roots define the inner polynomials p_{in} and ϕ_{in}. Next, solve the first right-hand side system to the requested relative residual tolerance with polynomial preconditioned GMRES, PP($d\phi_{\text{in}}$)-GMRES. Find the roots of the resulting outer GMRES residual polynomial, to define the outer polynomial p_{out}.
2. **Other right-hand side systems.** For the systems $Ax^j = b^j$, $j = 2 : nrhs$, compute $x^j = p_{\text{in}}(A)p_{\text{out}}(\phi_{\text{in}}(A))b^j$ using [26, Algorithm 1] for $\phi_{\text{in}}(A)$ and [26, Algorithm 3] for both p_{in} and p_{out}.

TABLE 1
Example 5: convection-diffusion matrix from $-u_{xx} - u_{yy} + 2u_x - 10^2 u = f$ of size $n = 40{,}000$ with 10 random right-hand sides. Compare using various $p(A) \approx A^{-1}$ to standard methods.

method	total MVP's	time
GMRES(100)	21.5 million	28.6 hours
BiCGStab	1.37 million	380 seconds
$p(A)$, deg = 1345	13,451	162 + 2.4 seconds
Double Polynomial deg = $40 \times 52 - 1 = 2079$	20,789	1.1 + 3.1 seconds
Nonsymmetric Lanczos deg = 1314	14,456	3.4 + 2.3 seconds

first system takes 162 seconds, while the next nine systems take only 2.4 seconds. The double polynomial generated with PP(40)-GMRES gives a polynomial of higher degree, 2079. (The outer ϕ polynomial is originally degree 50 and two stability roots are added, giving the overall degree $40 \times 52 - 1 = 2079$.) This method takes only 1.1 seconds to solve the first system and generate the approximation to A^{-1}; the following nine systems are solved in an additional 3.1 seconds, all reaching residual norm of 9.0×10^{-10} or better. For the last line of the table we use nonsymmetric Lanczos to construct p. The method is also very quick (more time for finding the polynomial, but even faster for solving the other systems, to similar accuracy).

Example 6. We generalize the last example to make the matrix increasingly indefinite: $-u_{xx} - u_{yy} + 2u_x - \gamma^2 u = f$ with increasing γ. For $\gamma = 8$, there are three negative eigenvalues, and the number increases with γ. For $\gamma = 10$, there are six negative eigenvalues, including two very close to zero: -7.93×10^{-6} and -7.96×10^{-6}. We will compare double polynomial approximations to A^{-1} to standard BiCGstab. PP-GMRES is run to 10^{-10} to generate the polynomial. The degree of the inner ϕ polynomial is increased with γ: it is 25 for $\gamma = 0$ and 8, and then goes up by 25 for each other case. BiCGStab is run with the limit of 100,000 iterations.

Table 2 shows the results. The double polynomial degree and runtime increase with indefiniteness (e.g., $\gamma = 100$ uses degree 40,774). The accuracy of using $p(A)b^j$ to solve systems $j = 2, \ldots, 10$ varies, but the results are fairly accurate for all cases, even with high degrees. The cost and accuracy of BiCGStab are about the same as the polynomial approach when $\gamma = 0$, but for $\gamma > 0$ BiCGStab is less competitive. Its cost increases, and the accuracy is sometimes considerably diminished.

Example 7. From the last example, we increase the nonnormality by increasing the convection coefficients in the differential equation: $-u_{xx} - u_{yy} + 25u_x + 10u_y - \gamma^2 u = f$. We change γ from 10 to 20, and see that the combination of nonnormality and indefiniteness makes the problem rather difficult.

Table 3 shows the results. BiCGStab does well for the definite case with $\gamma = 10$, then increasingly struggles. With $\gamma = 15$, all systems except one are solved to the requested tolerance of 10^{-8}, while for $\gamma = 18$, six are below 10^{-6} and finally for $\gamma = 20$ only three systems get to residual norm below 10^{-6} and the others are all above 10^{-2}. Using the double polynomial from polynomial preconditioned GMRES to solve the multiple right-hand sides is better, although the accuracy degrades as the indefiniteness increases. For the worst case of $\gamma = 20$, the degree of the inner polynomial is set to $d\phi_{\text{in}} = 175$, because this gives somewhat accurate results; other

TABLE 2
Example 6: convection-diffusion matrix ($n = 40{,}000$) from $-u_{xx} - u_{yy} + 2u_x - \gamma^2 u = f$ with 10 random right-hand sides. Compare a double polynomial to BiCGStab for increasing γ.

	Double Polynomial			BiCGStab		
γ	degree of polynomial	max res. norm	time (sec)	average MVP's	max res. norm	time (sec)
0	$25 \times 40 - 1 = 999$	3.0×10^{-10}	2.1	887	9.6×10^{-9}	3.0
8	$25 \times 60 - 1 = 1499$	2.2×10^{-9}	3.1	12,824	1.0×10^{-8}	39
10	$50 \times 41 - 1 = 2049$	6.5×10^{-9}	4.0	137,466	3.5×10^{-4}	463
20	$75 \times 96 - 1 = 7199$	5.4×10^{-11}	13	84,759	9.3×10^{-6}	296
35	$101 \times 154 - 1 = 15,554$	1.4×10^{-7}	29	97,843	5.2×10^{-8}	318
50	$126 \times 151 - 1 = 19,025$	6.1×10^{-10}	34	55,252	1.3×10^{-8}	189
75	$150 \times 195 - 1 = 29,249$	1.7×10^{-8}	54	119,789	9.9×10^{-9}	410
100	$175 \times 233 - 1 = 40,774$	8.2×10^{-8}	75	144,963	9.3×10^{-5}	497

TABLE 3
Example 7: convection-diffusion matrix ($n = 40{,}000$) from $-u_{xx} - u_{yy} + 25u_x + 10u_y - \gamma^2 u = f$ with 10 random right-hand sides. Compare double polynomial to BiCGStab for several γ.

	Double Polynomial			BiCGStab		
γ	degree of polynomial	max res. norm	time (sec)	average MVP's	max res. norm	time (sec)
10	$25 \times 42 - 1 = 1049$	8.7×10^{-10}	2.4	907	1.0×10^{-8}	3.0
15	$25 \times 73 - 1 = 1824$	4.7×10^{-7}	4.0	4426	2.8×10^{-2}	15
18	$50 \times 70 - 1 = 3499$	3.6×10^{-6}	8.3	53,714	6.5×10^{-2}	191
20	$175 \times 90 - 1 = 15,749$	8.8×10^{-5}	30	27,186	6.9×10^{-2}	97

values of $d\phi_{\text{in}}$ that we tried did not work as well. One can do better by using a lower degree polynomial and applying it more. For an inner polynomial of degree 50, a degree $50 \times 92 - 1$ polynomial gives solutions with a worst-case residual norm of 8.3×10^{-3}. But then applying it two more times, the accuracy improves to 4.8×10^{-9} and the total time is 23 seconds. This result suggests possibly trying to automate the number of times the polynomial is applied for difficult problems, which is not further pursued here. However, in the upcoming subsection on deflated polynomials, applying a polynomial several times to help with the deflation will be considered.

We finish this section with a theorem about accuracy of multiple right-hand side systems for normal A, using the distribution of the right-hand sides in the eigenvectors.

THEOREM 5. *Let $A \in \mathbb{C}^{n \times n}$ be a normal matrix, and let $Ax^{(1)} = b^{(1)}$ and $Ax^{(2)} = b^{(2)}$ be two systems with unit norm right-hand sides. Let A have the unitary diagonalization $A = Z\Lambda Z^*$, with eigenvectors $z_1, \ldots, z_n$ and corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$. Expand the right-hand sides as $b^{(1)} = Z(Z^* b^{(1)}) = \sum \beta_i^{(1)} z_i$ and $b^{(2)} = Z(Z^* b^{(2)}) = \sum \beta_i^{(2)} z_i$. Let the first residual vector be $r^{(1)} = b^{(1)} - A\widehat{x}^{(1)}$, where $\widehat{x}^{(1)}$ is the GMRES approximate solution with $\widehat{x}^{(1)} = p(A)b^{(1)}$. Obtain the second residual by premultiplying this $p(A)$ generated from the first linear system against the second right-hand side: $r^{(2)} = b^{(2)} - A\widehat{x}^{(2)}$, where $\widehat{x}^{(2)} = p(A)b^{(2)}$. Then*

$$\|r^{(2)}\| \leq \max \left| \frac{\beta_i^{(2)}}{\beta_i^{(1)}} \right| \cdot \|r^{(1)}\|.$$

Proof. Using that $r^{(2)} = b^{(2)} - A\widehat{x}^{(2)} = b^{(2)} - Ap(A)b^{(2)} = \pi(A)b^{(2)}$ and the eigenvalue decomposition of $\pi(A)$,

$$r^{(2)} = \pi(A)b^{(2)} = \sum_{i=1}^{n} \pi(\lambda_i) z_i z_i^* b^{(2)} = \sum_{i=1}^{n} \beta_i^{(2)} \pi(\lambda_i) z_i. \tag{8}$$

Since the eigenvectors form an orthonormal basis,

$$\begin{aligned} \|r^{(2)}\|^2 \;=\; \Big\| \sum_{i=1}^{n} \beta_i^{(2)} \pi(\lambda_i) z_i \Big\|^2 = \sum_{i=1}^{n} \big|\beta_i^{(2)} \pi(\lambda_i)\big|^2 &= \sum_{i=1}^{n} \Big| \frac{\beta_i^{(2)}}{\beta_i^{(1)}} \cdot \beta_i^{(1)} \pi(\lambda_i) \Big|^2 \\ &\le \max \left| \frac{\beta_i^{(2)}}{\beta_i^{(1)}} \right|^2 \sum_{i=1}^{n} \big|\beta_i^{(1)} \pi(\lambda_i)\big|^2 \\ &= \max \left| \frac{\beta_i^{(2)}}{\beta_i^{(1)}} \right|^2 \cdot \|r^{(1)}\|^2. \end{aligned}$$

∎

6. Deflated Polynomials for Solving Multiple Right-hand Sides. Small magnitude eigenvalues slow the convergence of iterative methods. After solving the first linear system with GMRES, one can use the resulting Krylov subspace to obtain approximations to the eigenvectors associated with the problematic small eigenvalues. These approximate eigenvectors can then be projected out from subsequent right-hand sides, greatly reducing the influence of the small eigenvalues and thus expediting convergence. This process, briefly mentioned in subsection 2.3, is called *deflation*. Here we explain how to integrate this idea with polynomial approximations to A^{-1} for lowering the required polynomial degree, and then solving subsequent linear systems.

Algorithm 5 describes three main steps for solving $Ax^j = b^j$ for $j = 1, \ldots, nrhs$. Step 1. Solve the first system using polynomial preconditioned GMRES. From the resulting outer subspace, approximate eigenvectors via a standard Rayleigh–Ritz procedure. Use these approximate eigenvectors for a Galerkin projection applied to the other right-hand sides, which (hopefully) deflates the most significant small eigenvalues from these systems. (See Algorithm 2 for details; the columns of the $n \times nev$ matrix V span the nev approximate eigenvectors we wish to deflate, giving the approximate solutions x_e^j for $j = 1, \ldots, nrhs$.) Step 2. Solve the deflated second system ($Ax^2 = b^2 - Ax_e^2$) with polynomial preconditioned GMRES using the same inner polynomial as before. This solve develops a double polynomial, which we call a *deflated polynomial* (since the small magnitude eigenvalues are deflated from this second system). If the deflation is effective, the polynomial degree is reduced. Step 3. Apply this deflated polynomial to solve the systems with all other (deflated) right-hand sides.

For a significantly nonnormal matrix, deflation is less effective unless both right and left eigenvectors are computed. For these cases, we will consider applying a lower degree polynomial more than once; see Example 9.

Example 8. We use the matrix BWM2000 from SuiteSparse [11]. Though not large ($n = 2000$) and only mildly indefinite (all eigenvalues have negative real parts, except for two that are barely positive), solving linear equations with this matrix can be difficult. The right-hand sides are again random unit vectors. The parameters are $d\phi_{in} = 50$, $nev = 30$, $rtol_1 = 10^{-11}$, $rtol_2 = 10^{-9}$, and $rtol_3 = 10^{-8}$. PP(50)-GMRES takes 64 iterations and finds 30 approximate eigenvectors with residual norms ranging from 6.5×10^{-8} to 4.7×10^{1}. Deflated PP-GMRES on the second system takes 16

Algorithm 5 A deflated polynomial for solving multiple right-hand sides

0. Preliminary. Choose $d\phi_{\text{in}}$, the degree of the inner polynomial ϕ_{in}. Pick *nev*, the number of eigenvalues to deflate. Choose relative residual norm tolerances for the three steps: the first system is solved to $rtol_1$, the second to $rtol_2$ and $rtol_3$ for the other systems.

1. Solve first system and compute approximate eigenvectors. Run GMRES for $d\phi_{\text{in}}$ iterations to develop the inner polynomials p_{in} and ϕ_{in}. Then solve the first right-hand side system with polynomial preconditioned GMRES, i.e., run PP($d\phi_{\text{in}}$)-GMRES with $rtol_1$. From the subspace thus developed, apply the Rayleigh–Ritz procedure to compute approximate eigenvectors corresponding to the *nev* smallest eigenvalues [13].

2. Develop the deflated polynomial with the second system. Deflate the second right-hand side system with a Galerkin projection over the approximate eigenvectors (Algorithm 2). Then apply PP($d\phi_{\text{in}}$)-GMRES, with the same inner polynomial developed earlier, to the deflated second system with $rtol_2$. This generates an outer polynomial p_{out} and thus the deflated double polynomial $p(z) = p_{\text{in}}(z)p_{\text{out}}(\phi_{\text{in}}(z))$.

3. Other systems. For $Ax^j = b^j$, $j = 3, \ldots, nrhs$, project over the approximate eigenvectors, yielding the partial solution x_e^j and deflated system $A(x^j - x_e^j) = r^j$. Next, the approximate solution to the original system is $x^j = x_e^j + p(A)r^j$.

4. Optional. Reapply the deflated polynomial to systems that have not converged to $rtol_3$, including the second system if more accuracy than $rtol_2$ is needed. Each reapplication needs a projection and multiplication by $p(A)$.

TABLE 4
Example 8: BWM2000 matrix of size $n = 2000$ with 10 right-hand sides. Compare a deflated polynomial for multiple right-hand sides to BiCGStab.

method	total MVP's	total time	log avg. res. norm
Deflated Double Poly deg $= 50 \times 16 - 1 = 799$	10,460	0.37 + 0.09 + 0.07 = 0.53 seconds	3.1×10^{-10}
BiCGStab	550,313	6.9 seconds	5.3×10^{-6}

iterations. Figure 10 shows the convergence of PP(50)-GMRES on the *first* system with a (blue) dashed line and BiCGStab on the same system with a (green) dash-dot line. BiCGStab converges only to residual norm 4.6×10^{-3} in 45,405 matrix-vector products (most not shown on the plot). The (red) solid line shows the solution of the deflated system corresponding to the *second* right-hand side, which proceeds much quicker due to the deflation.

Table 4 shows the cost for solving 10 right-hand sides. The deflated double polynomial method uses 0.37 seconds to solve the first system and compute approximate eigenvectors. Then 0.09 seconds are for the second system and generating a deflated polynomial. The other systems take 0.07 seconds, and all solutions have residual norms of 8.3×10^{-10} or smaller. Meanwhile, BiCGStab is applied to the ten systems with tolerance of 10^{-8} and a maximum of 100,000 iterations. This process takes 6.9 seconds, but only three of the systems converge. The log average for the systems (exponential of the average of the natural logs of the residual norms) is 5.3×10^{-6}.

Example 9. Consider the one-dimensional convection-diffusion equation $-u'' + \alpha u' - 30^2 u = f$. This example shows that it can take careful implementation of

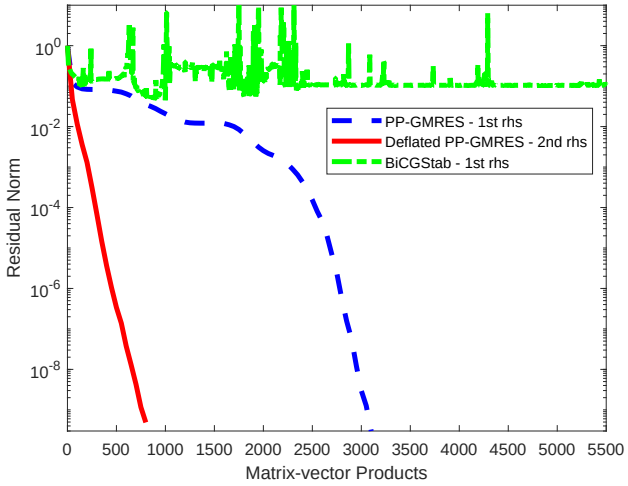

FIG. 10. *Example 8: the matrix is BWM2000. Converge is shown for BiCGStab, polynomial preconditioned PP(50)-GMRES with no restarting, and for deflated PP(50)-GMRES.*

TABLE 5
Example 9: one-dimensional convection-diffusion equation $-u'' + \alpha u' - 30^2 u = f$ *with* $n = 1000$.

	Double Polynomial			BiCGStab	
α	rtol's ($\log_{10}$)	degree of deflated polynomial	average MVP's	average MVP's	log avg. res. norm
0	$-11,-9,-8$	$25 \times 10 - 1 = 249$	392	57,868	6.6×10^{-9}
5	$-11,-9,-8$	$25 \times 62 - 1 = 1549$	1562	61,623	2.9×10^{-7}
5	$-11,-3,-8$	$25 \times 4 - 1 = 99$	437		
25	$-10,-2,-8$	$25 \times 3 - 1 = 74$	542	66,275	1.4×10^{-3}

the deflated polynomial for significantly nonnormal and indefinite problems, but the method can be cheap and accurate. Table 5 shows results for $\alpha = 0, 5, 25$. With $n = 1000$, the matrix is fairly ill-conditioned and the u term makes these matrices indefinite. For the symmetric case of $\alpha = 0$, the first system is solved with PP(25)-GMRES to relative residual tolerance of 10^{-11}, requiring 64 outer iterations and 1625 matrix-vector products. Then 30 eigenvectors are computed. Next, the deflated polynomial is found of degree 249, and it accurately solves the other right-hand sides. The residual norms are 6.9×10^{-9} or better, and the average number of matrix-vector products (including the first right-hand side) is 387. BiCGStab needs over two orders of magnitude more matrix-vector products, but it does produce accurate results.

Next, consider $\alpha > 0$. BiCGStab requires slightly more matrix-vector products, and is less accurate. The deflated polynomial is not effective for $\alpha = 5$: it has about the same degree as if no deflation was used, due to the inaccuracy of the deflation in removing eigenvectors. PP-GMRES converges rapidly at first due to the partial deflation, but slows down once the eigenvector components of the residual vector are reduced down to the level of the deflated components. At that point, the Krylov method must deal with these small eigenvalues, and the convergence plateaus while that happens; see the solid (red) curve in Figure 11. For effective use of the deflation, it pays to stop the PP-GMRES method when the approximately deflated eigenvalues begin to impede convergence. Doing so gives a low degree deflated polynomial that

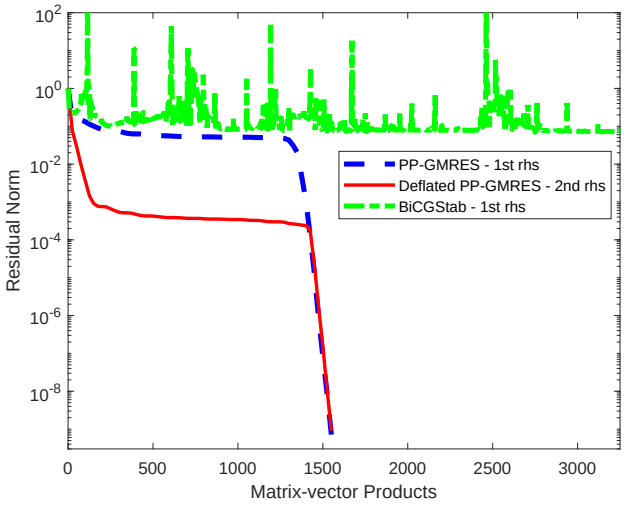

FIG. 11. *Example 9: Convection-diffusion equation* $-u'' + 5u' - 30^2 u = f$, $n = 1000$. *Convergence is shown for BiCGStab, PP(25)-GMRES with no restarting, and for deflated PP(25)-GMRES.*

needs to be used more than once, with a deflation in-between each application of the polynomial. Table 5 shows the results of running PP(25)-GMRES on the second system to $rtol_2 = 10^{-3}$, then applying it three times (with projections) to b^j for $j = 3, \ldots, 10$ and twice for the partly solved right-hand side for $j = 2$. This approach gives accurate results, with a log average residual norm for the nine right-hand sides of 5.8×10^{-10}. The average matrix-vector products for all 10 systems is only 427.

For higher nonnormality, $\alpha = 25$, a good result comes from $rtol_2 = 10^{-2}$. The deflated polynomial is applied five or six times for $j = 2, \ldots, 10$ to reach the desired residual norm level of 10^{-8}. While this uses two orders of magnitude fewer matrix-vector products than BiCGStab and gives better accuracy, it required experimentation. An automated method for choosing $rtol_2$ would be quite desirable.

Example 10. In Lattice Quantum Chromodynamics (QCD), one important task is to estimate the trace of the inverse of a large non-Hermitian complex matrix. This can be done with a Monte Carlo approach called Hutchinson's method [18], which requires solving many linear systems with random right-hand sides. Here we use 10 right-hand sides with entries randomly drawn from $\{\pm 1, \pm \mathrm{i}\}$. The QCD matrix is from a 16^3-by-24 lattice, giving $n = 1{,}179{,}648$. The desired relative residual tolerance for the linear systems is $rtol_3 = 10^{-5}$. The spectrum is roughly inside a circle in the right-half of the complex plane, with eigenvalues approaching the origin from above and below; the presence of small eigenvalues makes deflation important. A two-sided projection is needed. Once right eigenvectors are known, the left eigenvectors can be computed easily for this application. Thus only one deflation projection is needed despite the significant nonnormality. This projection is the same as the Galerkin projection in Algorithm 2 except in Step 1, we compute a matrix W whose columns span the subspace of left eigenvectors and in Step 2, $H = W^*AV$ and $c = W^*r_0$.

Table 6 compares the deflated polynomial to the non-deflated polynomial and BiCGStab. The non-deflated polynomial is found by solving the first right-hand side with PP(40)-GMRES to relative residual norm accuracy below 10^{-6}. This requires 52 iterations, so the polynomial is of high degree $40 \times 52 - 1 = 2081$. This polynomial solves the next nine systems to accuracy 2.4×10^{-5} or better, which is just worse than

TABLE 6
Example 10: QCD matrix of size n = 1,179,648 with 10 right-hand sides. Comparison of a deflated polynomial, non-deflated polynomial, and the conventional BiCGstab algorithm.

method	total MVP's	total time
Deflated Double Polynomial deg = $40 \times 14 - 1 = 799$	10,524	7.1 + 1.1 + 8.5 = 16.7 minutes
Non-deflated Double Polynomial deg = $40 \times 52 - 1 = 2081$	20,898	4.1 + 31.7 = 35.8 minutes
BiCGStab	203,481	375 minutes

the desired accuracy; a slightly higher degree polynomial is needed. Next, BiCGStab is solved to requested relative residual of 10^{-5} and gives accurate answers, but requires much more time.

Finally, to find the deflated polynomial, the first right-hand side is solved with PP(40)-GMRES to relative residual tolerance of $rtol_1 = 10^{-12}$, which takes 78 iterations. Then approximate eigenvectors are computed and here only the $nev = 34$ of them that have residual norm below 10^{-3} are used, along with the associated left eigenvectors. Deflated PP(40)-GMRES for the second right-hand side with $rtol_2 = 10^{-6}$ runs for only 14 iterations. It generates a polynomial of degree $40 \times 14 - 1 = 559$, much lower than the non-deflated case. This polynomial solves the remaining eight systems to relative residual norms of 3.4×10^{-6} or better. Overall, the time is reduced by more than a factor of two compared to the non-deflated polynomial. More time is needed for the first right-hand side (7.1 minutes versus 4.1 minutes), since it is over-solved. However, the next nine are faster (9.6 minutes instead of 31.7 minutes).

7. Stability of the polynomial. Polynomials from GMRES are especially prone to instability when a few eigenvalues stand out from the rest of the spectrum. Stability can be improved by adding extra copies of the harmonic Ritz values near these outstanding eigenvalues; see Subsection 2.2. We briefly consider implications for polynomials approximating A^{-1}. When high-degree polynomials are needed, stability is of particular concern. The next example shows that it can be possible to find an effective polynomial even when there are very outstanding eigenvalues.

Example 11. We use bidiagonal matrices with increasing separation of the large eigenvalues and thus increasing difficulty for stability. All matrices are size $n = 2500$ and have 0.2's on the superdiagonal. The first matrix has eigenvalues $1, 2, 3, \ldots, n$ on the main diagonal. Matrix 2 has diagonal $0.1, 0.2, 0.3, \ldots, 0.9, 1, 2, 3, \ldots, 2490, 2491$. While the large eigenvalues are not especially separated, they are more separated relative to the size of the small eigenvalues than for the first matrix. Matrix 3 has eigenvalues $0.1, 0.2, 0.3, \ldots, 0.9, 1, 2, 3, \ldots, 2490, 2600$, so there is one very well separated eigenvalue. Matrix 4 has five outstanding eigenvalues, with diagonal $0.1, 0.2, 0.3, \ldots, 0.9, 1, 2, 3, \ldots, 2486, 2600, 2700, 2800, 2900, 3000$.

Table 7 shows results both without and with the stability control of adding roots from Algorithm 1. The first right-hand side is solved to relative residual tolerance of 10^{-11}, and the polynomial p is then applied to nine other right-hand sides. Without adding roots, the polynomial is effective in solving multiple right-hand sides only for Matrix 1. In that case, the nine extra systems all have residual norms below 3.1×10^{-11}. For the next three matrices, the polynomial is increasingly unstable with high *pof* values (as defined in Algorithm 1) that indicate steep slopes, and it provides very inaccurate solutions for the nine additional right-hand sides.

TABLE 7
Example 11: Bidiagonal matrices with some increasingly outlying eigenvalues. The effect of stability control is shown for solving systems with multiple right-hand sides.

	Without Stability Control			With Stability Control	
matrix	degree of polynomial	max *pof*	max residual norm	roots added	max residual norm
1	324	2.5×10^{1}	3.1×10^{-11}	0	3.1×10^{-11}
2	596	9.5×10^{22}	5.4×10^{6}	12	2.7×10^{-11}
3	596	$2,3 \times 10^{103}$	5.3×10^{87}	19	5.7×10^{-9}
4	597	7.9×10^{216}	4.5×10^{201}	68	1.5×10^{-11}

With stability control from Algorithm 1 using *pofcutoff* $= 8$, results are good even with outlying eigenvalues and many added roots. Matrix 3 gives the least accurate results, however if *pofcutoff* $= 4$, then all nine residual norms are below 2.3×10^{-11}.

The improvement with stability control is remarkable. However, the next example is designed to trouble the stability control procedure.

Example 12. We choose a diagonal matrix with entries $0.1, 0.2, 0.3, \ldots, 1, 2, 3, \ldots, 50, 551, 552, 553, \ldots 1000, 1501, 1502, 1503, \ldots, 2000, 2501, 2502, 2503, \ldots, 3000, 3501, 3502, 3503, \ldots, 4491$. This spectrum has four prominent gaps, and the residual polynomial π has steep slope at eigenvalues at their edges. Also contributing to the steepness is that there are enough small eigenvalues to push the polynomial degree high. Slopes are especially steep for the eigenvalue at 50 and the ones just below it.

We solve the first system to relative residual norm below 10^{-11}, generating p of degree 454. Without stability control, this polynomial does not accurately solve other systems: the residual norms for nine right-hand sides are as high as 9.4×10^{-3}. With stability control, the results are even *worse*: the largest of the nine residual norms is 3.2×10^{-1}. Seven roots are added, including extra roots near eigenvalues 50, 49, 48, 47 and 46, along with two at Ritz values in gaps of the spectrum. Figure 12 (top) shows the polynomial ϕ before roots are added (dashed line) and with roots (solid line). This polynomial needs to be close to 1 over the spectrum for the method to be effective. Figure 12 (bottom) examines these polynomials over the small and large eigenvalues. Adding five roots near 50 multiplies the π polynomial by linear factors like $(1 - z/50)$, each of which is near -100 at the large eigenvalues. The polynomial oscillates with larger amplitude around the large eigenvalues, causing two problems: instability due to a *pof* of 5×10^{13} at the largest eigenvalue, and, more importantly, the polynomial ϕ varies significantly from 1 at quite a few of the eigenvalues (e.g., 4472, 4473 and 4474). This inaccuracy makes $p(A)$ useless for solving linear equations.

We propose an improved version of Algorithm 1 that performs the stability test on roots in increasing order of magnitude, and then updates the *pof* values when a root is added; see Algorithm 6. This algorithm improves the results so that the maximum of the nine residual norms is 4.0×10^{-6}. The limited accuracy is not due to instability, but rather the ϕ polynomial is still not close enough to 1 at some eigenvalues. The largest deviation is at $\lambda_{1512} = 3503$ where $\phi(\lambda_{1512}) = 1 - 5.2 \times 10^{-5}$, which is significantly far from 1. The ϕ polynomial is more accurate at larger eigenvalues due to two added roots at the largest two eigenvalues, a result of the updated *pof* values. This example demonstrates the danger of adding roots to the GMRES polynomial, because then the polynomials no longer correspond to a minimum residual method and may not be accurate at all eigenvalues. However, Algorithm 6 with *pof* updating did improve

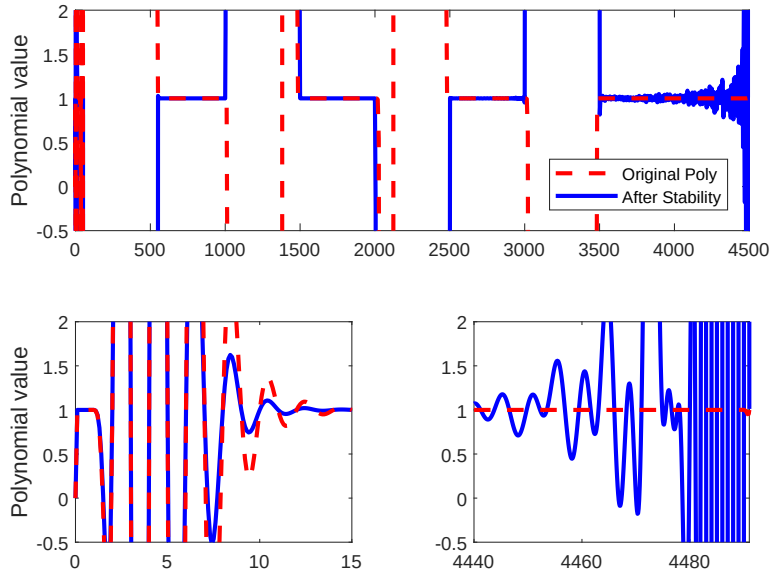

FIG. 12. *Example 12: a matrix with four gaps in the spectrum, $n = 2500$. The top plot shows the ϕ polynomial of degree 455, both before and after roots are added for stability control. The bottom plot shows a close-up around the small and large parts of the spectrum.*

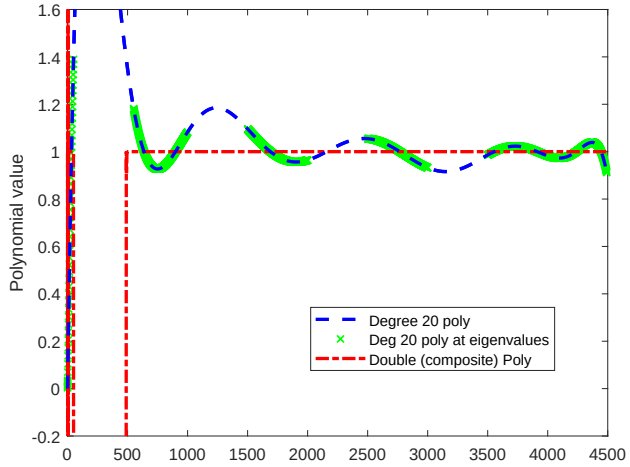

FIG. 13. *Example 12: a matrix with four gaps in the spectrum, $n = 2500$. The plots are for a double polynomial. The inner polynomial ϕ_{in} is highlighted at the eigenvalues. Also plotted is the double polynomial $\phi_{\text{out}}(\phi_{\text{in}}(z)) = p_{\text{in}}(z) * p_{\text{out}}(\phi_{\text{in}}(z))$ of degree 899.*

accuracy. In other testing with matrices that are difficult due to gaps in the spectrum or due to indefiniteness, Algorithm 6 often improves the situation.

We now describe another way to improve stability through use of the double polynomial. For the same matrix with gaps, we run PP(20)-GMRES to relative residual norm below 10^{-11}, which takes 45 iterations; this process yields the double polynomial. No extra roots are needed for stability for either the inner or outer parts of the double polynomial. The p polynomial is degree $20 \times 45 - 1 = 899$, much higher than the polynomial from GMRES. Applying this polynomial to the nine extra right-hand sides gives residual norms at or below 3.9×10^{-10}. Figure 13 shows

Algorithm 6 Adding Roots to $\pi(z)$ for Stability with *pof* Updating

1. **Setup:** Assume the d roots $(\theta_1, \ldots, \theta_d)$ of π have been computed and then sorted according to the modified Leja ordering [4, alg. 3.1]. (For high-degree polynomials and/or large magnitude roots, use sums of logs in place of products, to prevent overflow and underflow in the Leja ordering calculations.)
2. **Compute** *pof*(k)**:** For $k = 1, \ldots, d$, compute $pof(k) = \prod_{i \neq k} |1 - \theta_k/\theta_i|$.
3. **Add roots using a reordered list of harmonic Ritz values and update** *pof* **values:** Reorder the roots and their corresponding *pof*'s (generally by increasing magnitude). For each root θ_k starting with the smallest, compute $\lceil \log_{10}(pof(k)) - pofcutoff)/14 \rceil$. Add that number of θ_k copies to the list of roots with the Leja ordering. Add the first to the end of the list; if there are others, space them evenly between the first and last occurrence of θ_k (keeping complex roots together). Based on the added root(s), update the *pof* values of all roots that have not yet been considered in the list of roots sorted by magnitude. Then continue checking the *pof* values for the next root.
4. **Apply a second Leja ordering:** For the list of θ_k values that is Leja ordered aside from the stabilizing roots, perform a second Leja ordering. To give distinct values for the Leja algorithm, perturb these roots slightly for determining the order, but do not change the actual θ_k values. (We perturb to $(1 + 10^{-12} * randn) * \theta_k$, where "*randn*" is a random Normal(0,1) number.)

the degree 20 inner polynomial ϕ_{in} as a dashed line and highlights the values at the eigenvalues. It also shows the composite ϕ polynomial, $\phi(z) = \phi_{\text{out}}(\phi_{\text{in}}(z))$, which is near 1 throughout the spectrum. This polynomial is also near 1 over most of the gaps in the spectrum. This is because the inner polynomial maps the gaps (except for the first one) into a zone where eigenvalues are also mapped, and so the outer polynomial needs to be near 1 there. The fact that the double polynomial is near 1 over most gaps, instead of being unconstrained, partly explains why the higher degree is needed.

Stability control and double polynomials enable polynomial approximation to the inverse for some difficult matrices, but these approaches are not foolproof. One can construct adversarial examples with enough small eigenvalues and gaps in the spectrum to make an effective polynomial quite hard to find. On the other hand, many applications have matrices that lack such challenges. Polynomial approximation can be effective for these problems, even when a high-degree polynomial is needed.

8. Conclusion. It is often possible to construct an accurate moderate-degree polynomial approximation to the inverse of a large matrix. The theory for the normal case shows that accuracy of the polynomial follows that of the GMRES residual, along with the interplay of the right-hand side and the eigenvectors. The approximating polynomial can be found in several ways; here we focused on full GMRES and polynomial preconditioned GMRES, the latter of which builds double (composite) polynomials. This composite approach can be more efficient due to reduced orthogonalization costs, though higher degree polynomials are often needed.

Applications for approximate polynomial inverses include solving systems of linear equations with multiple right-hand sides. For difficult problems, the polynomial is often better than BiCGStab in expense and accuracy. However, like all Krylov methods, a polynomial is not always effective. This is particularly true for matrices that are ill-conditioned and have outstanding eigenvalues. Stability control can help.

A deflated version of the polynomial uses approximate eigenvectors to lower the

degree. Multiple applications of the deflated polynomial may be needed, particularly for significantly nonnormal matrices. Some fine-tuning is needed.

In future research, we plan to consider both the nonsymmetric and symmetric Lanczos algorithms. These methods have some natural stability control through roundoff error, which produce extra copies of Ritz values corresponding to outstanding eigenvalues (so-called "ghost" Ritz values). Nonsymmetric Lanczos can find both right and left eigenvectors, which could help construct polynomials for deflation; the stability of this process will require careful assessment.

REFERENCES

[1] A. M. Abdel-Rehim, R. B. Morgan, and W. Wilcox, *Improved seed methods for symmetric positive definite linear equations with multiple right-hand sides*, Numer. Linear Algebra Appl., 21 (2014), pp. 453–471.

[2] S. F. Ashby, *Polynomial preconditioning for conjugate gradient methods*. PhD Thesis, University of Illinois at Urbana-Champaign, 1987.

[3] S. F. Ashby, T. A. Manteuffel, and J. S. Otto, *A comparison of adaptive Chebyshev and least squares polynomial preconditioning for conjugate gradient methods*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1–29.

[4] Z. Bai, D. Hu, and L. Reichel, *A Newton basis GMRES implementation*, IMA J. Numer. Anal., 14 (1994), pp. 563–581.

[5] A. H. Baker, E. R. Jessup, and T. Manteuffel, *A technique for accelerating the convergence of restarted GMRES*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 962–984.

[6] C. Beattie, *Harmonic Ritz and Lehmann bounds*, Electron. Trans. Numer. Anal., 7 (1998), pp. 18–39.

[7] P. N. Brown, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.

[8] Z.-H. Cao, *A note on the convergence behavior of GMRES*, Appl. Num. Math., 25 (1997), pp. 13–20.

[9] R. L. Carden and M. Embree, *Ritz value localization for non-Hermitian matrices*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1320–1338.

[10] M. Crouzeix and C. Palencia, *The numerical range is a* $(1+\sqrt{2})$*-spectral set*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 649–655.

[11] T. A. Davis, *Suitesparse sparse matrix collection*, 2024. https://sparse.tamu.edu/.

[12] M. Embree, *How descriptive are GMRES convergence bounds?* arXiv:2209.01231, 2022.

[13] M. Embree, J. A. Loe, and R. B. Morgan, *Polynomial preconditioned Arnoldi with stability control*, SIAM J. Sci. Comput., 43 (2021), pp. A1–A25.

[14] V. Faber, W. Joubert, E. Knill, and T. Manteuffel, *Minimal residual method stronger than polynomial preconditioning*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 707–729.

[15] B. Fischer and L. Reichel, *A stable Richardson iteration method for complex linear systems*, Numer. Math., 54 (1988), pp. 225–241.

[16] R. W. Freund, *Quasi-kernel polynomials and their use in non-Hermitian matrix iterations*, J. Comput. Appl. Math., 43 (1992), pp. 135–158.

[17] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.

[18] M. F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Commun. Stat. Simul. Comput., 18 (1989), pp. 1059–1076.

[19] W. Joubert, *A robust GMRES-based adaptive polynomial preconditioning algorithm for nonsymmetric linear systems*, SIAM J. Sci. Comput., 15 (1994), pp. 427–439.

[20] S. A. Kharchenko and A. Y. Yeremin, *Eigenvalue translation based preconditioners for the GMRES(k) method*, Numer. Linear Algebra Appl., 2 (1995), pp. 51–77.

[21] C. Lanczos, *Chebyshev polynomials in the solution large-scale linear systems*, Proc. ACM, (1952), pp. 124–133.

[22] P. Lashomb, R. B. Morgan, T. Whyte, and W. Wilcox, *High-degree polynomial noise subtraction for disconnected loops*, Comput. Phys. Commun., 298 (2024), pp. 109120–1 – 109120–7.

[23] ———, *Multi-polynomial Monte Carlo for QCD trace estimation*, Comput. Phys. Commun., 300 (2024), pp. 109163–1 – 109163–10.

[24] R. Li, Y. Xi, E. Vecharynski, C. Yang, and Y. Saad, *A thick-restart Lanczos algorithm with polynomial filtering for Hermitian eigenvalue problems*, SIAM J. Sci. Comput., 38 (2016),

pp. A2512–A2534.
[25] Q. Liu, R. B. Morgan, and W. Wilcox, *Polynomial preconditioned GMRES and GMRES-DR*, SIAM J. Sci. Comput., 37 (2015), pp. S407–S428.
[26] J. A. Loe and R. B. Morgan, *Toward efficient polynomial preconditioning for GMRES*, Numer. Linear Algebra Appl., 29 (2021), pp. 1–21.
[27] T. A. Manteuffel and J. S. Otto, *On the roots of the orthogonal polynomials and residual polynomials associated with a conjugate gradient algorithm*, Numer. Linear Algebra Appl., 1 (1994), pp. 449–475.
[28] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Algebra Appl., 154–156 (1991), pp. 289–309.
[29] ———, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[30] ———, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.
[31] R. B. Morgan and W. Wilcox, *Deflated iterative methods for linear equations with multiple right-hand sides.* arXiv:math-ph/0405053v2, 2004.
[32] R. B. Morgan and M. Zeng, *Harmonic projection methods for large non-symmetric eigenvalue problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 33–55.
[33] N. M. Nachtigal, L. Reichel, and L. N. Trefethen, *A hybrid GMRES algorithm for non-symmetric linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 796–825.
[34] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–133.
[35] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.
[36] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, SIAM Classics ed., 1998.
[37] H. Rutishauser, *Theory of gradient methods*, in Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems, M. Engeli, T. Ginsburg, H. Rutishauser, and E. Stiefel, eds., Birkhauser, Basel, 1959, pp. 24–49.
[38] Y. Saad, *Chebychev acceleration techniques for solving large nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567–588.
[39] ———, *Least squares polynomials in the complex plane and their use for solving sparse nonsymmetric linear systems*, SIAM J. Numer. Anal., 24 (1987), pp. 155–169.
[40] ———, *Iterative Methods for Sparse Linear Systems, 2nd Edition*, SIAM, Philadelphia, PA, 2003.
[41] Y. Saad and M. H. Schultz, *GMRES: a generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[42] J. A. Sifuentes, M. Embree, and R. B. Morgan, *GMRES convergence for perturbed coefficient matrices, with application to approximate deflation preconditioning*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1066–1088.
[43] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM J. Sci. Comput., 16 (1995), pp. 917–933.
[44] D. C. Smolarski and P. E. Saylor, *An optimal iterative method for solving any linear system with a square matrix*, BIT, 28 (1988), pp. 163–178.
[45] A. Stathopoulos and K. Orginos, *Computing and deflating eigenvalues while solving multiple right hand side linear systems in quantum chromodynamics*, SIAM J. Sci. Comput., 32 (2010), pp. 439–462.
[46] E. L. Stiefel, *Kernel polynomials in linear algebra and their numerical applications*, Nat. Bur. Standards, Appl. Math. Ser., 49 (1958), pp. 1–22.
[47] H. K. Thornquist, *Fixed-Polynomial Approximate Spectral Transformations for Preconditioning the Eigenvalue Problem*, PhD thesis, Rice University, 2006. Technical report TR06-05, Department of Computational and Applied Mathematics, Rice University.
[48] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*, Princeton Univ. Press, Princeton, NJ, 2005.
[49] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 631–644.
[50] M. B. van Gijzen, *A polynomial preconditioner for the GMRES algorithm*, J. Comput. Appl. Math., 59 (1995), pp. 91–107.
[51] X. Ye, Y. Xi, and Y. Saad, *Proxy-GMRES: Preconditioning via GMRES in polynomial space*, SIAM J. Sci. Comput., 42 (2021), pp. 1248–1267.
[52] B. Zhong and R. B. Morgan, *Complementary cycles of restarted GMRES*, Numer. Linear Algebra Appl., 15 (2008), pp. 559–571.