Analysis of Polynomial Preconditioners

Ferdinand Vanmaele
Universität Heidelberg
Email: Vanmaele@stud.uni-heidelberg.de

***Abstract*—We explore solving large systems of linear equations using the GMRES method, and study the effect of polynomial preconditiong for this method. Besides improving convergence for difficult problems, polynomial preconditioning allows us to reduce the amount of required scalar products, at the cost of more SpMV (sparse matrix-vector multiplication) operations. Preconditioning thus improves performance when solving such systems on massively parallel systems, where communication costs of scalar products can be a bottleneck.**

***Index Terms*—Krylov subspaces, GMRES, polynomial preconditioning, communication-avoiding**

I. Introduction

Systems of linear equations $Ax = b$, where A is a sparse matrix of high dimension n, are widely solved in scientific computing through iterative methods, that is, by finding a succession of approximations from an initial value [3, Chapter 6]. This is in contrast to direct methods, which are based on the initial factorization of the coefficient matrix and give the result in a number of exactly predictable operations. The disadvantage of direct methods lies in a high number of arithmetic operations, and in the memory footprint of storing matrices that were initially sparse [4, Example 3.3].

GMRES (Generalized Minimum Residual) method is an iterative method suitable for a wide range of linear systems, including indefinite and nonsymmetric systems. In every step, it minimizes the norm of the residual vector $\|b - A\tilde{x}\|$ over a Krylov subspace. Constructing the Krylov subspace is done through a Gram-Schmidt orthogonalization procedure using inner products. On parallel and distributed systems, computing inner products requires global reduction operations involving all processes. In comparison, sparse matrix-vector multiplications only need to exchange data between neighbors [3, 14.5].

To address this issue, we consider preconditioned systems $M^{-1}Ax = M^{-1}b$, where M^{-1} is a (typically low degree) matrix polynomial. This preconditioned system has the same solution as the original system $Ax = b$, but may achieve faster convergence. Achieving a certain tolerance for the residual thus occurs in less GMRES steps (inner products), at the expense of more matrix-vector multiplications in every step. We discuss a polynomial which does not require knowledge of the spectrum of the matrix A. Finding the coefficients of M^{-1} requires a (possibly dense) matrix-matrix multiplication. This cost can be controlled by the degree of the polynomial, depending on the difficulty of the problem, making it more granular than classical preconditioners such as Incomplete LU (ILU) factorizations.

II. Related Work

Saad [1] introduced the GMRES method for systems in which the coefficent matrix A is nonsymmetric or indefinite. Indefinite systems commonly arise, for example, in mixed finite element methods for solving problems in fluid and solid mechanics [6], [7].

Since then, the algorithms have been included in general textbooks on numerical linear algebra and scientific computing [4], [5], [3]. Implementations are available in MATLAB and scientific computing libraries such as GSL and TRILINOS. Liu [9] discusses polynomial preconditioning with GMRES and introduces a way to find the coefficients without knowledge of the matrix spectrum. Loe et. al. [10] conducted further experiments on (shared-memory) parallel hardware with TRILINOS.

Besides polynomial preconditioning, Block Krylov methods also help reduce the effect of inner products on communication. Saad [4, 6.12] put forward an overview and gives further references.

After revisiting theoretical results on Krylov subspaces and GMRES in Sections III, IV and V, we introduce polynomial preconditioning and its advantages in Section VI. In Section VII we test the impact of polynomial preconditioning on convergence for several test problems.

III. Krylov subspaces

We will look at iterative methods for solving the linear system $Ax = b$ of dimension n, where A is an invertible square matrix, with approximate solutions in Krylov subspaces. These methods can be applied to either real or complex matrices. We introduce Krylov subspaces and construct an orthonormal basis. Solving a least squares problem to find the minimum residual for vectors in this subspace leads to the GMRES method.

Definition 1 ([4, 6.2])**.** The **Krylov subspace** of order m, denoted by $\mathcal{K}_p$, is the vector space spanned by $v \in \mathbb{R}^n$ and the first $m-1$ powers of A:

$$\mathcal{K}_m(A, v) = \text{span}\{v, Av, A^2v, \ldots, A^{m-1}v\}.$$

The Krylov subspaces form an increasing family of subspaces, with dimension bounded by n. It is the subspace of all vectors in $\mathbb{R}^n$ which can be written as $x = p(A)v$, where p is a polynomial of degree not exceeding $m-1$.

In practice, constructing the Krylov subspaces amounts to determining their basis. The power basis $(v, Av, A^2v, \ldots, A^{m-1}v)$ converges to (a multiple of) an eigenvector that corresponds to the largest eigenvalue of

A, in absolute value [3, 10.2]. Therefore we construct an orthonormal basis $\{v_1, v_2, \dots, v_m\}$ as follows.

Arnoldi's method applies the (modified) Gram-Schmidt orthonormalization procedure to the vectors obtained by successive products of the matrix A. Algorithm 1 [4, 6.3.2] gives a practical implementation.

Algorithm 1 Arnoldi-Modified Gram-Schmidt

Choose a vector v_1 of norm 1
For $j = 1, 2, \dots, m$ **Do**:
  Compute $w_j := Av_j$
  For $i = 1, \cdots, j$ **Do**:
    $h_{ij} = (w_j, v_i)$
    $w_j := w_j - h_{ij}v_i$
  EndDo
  $h_{j+1,j} = \|w_j\|_2$. If $h_{j+1,j} = 0$ **Stop**
  $v_{j+1} = w_j/h_{j+1,j}$
EndDo

The modified Gram-Schmidt procedure has better numerical properties than the classical procedure [4, 6.3.2], but requires additional synchronization points when implemented on a parallel computer. Additional methods such as DCGS2 were introduced to address these issues [11]. Householder transformations, which are relatively costly but stable, were also proposed [8, 2.3.4].

The following important relations hold for the Arnoldi basis:

$$AV_m = V_{m+1}\overline{H}_m, \qquad V_m^T A V_m = H_m,$$

where V_m denotes the $n \times m$ matrix with orthonormal column vectors $v_1, \dots, v_m$, $\overline{H}_m$ the $(m+1) \times m$ Hessenberg matrix with nonzero entries h_{ij}, and H_m the matrix obtained from $\overline{H}_m$ by deleting its last row: [4, Proposition 6.5]

$$H_m = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & & & \vdots \\ & h_{3,2} & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \end{pmatrix}.$$

IV. The GMRES method

Let $r_0 = b - Ax_0$ denote the residual for an initial approximation x_0. Let $\mathcal{K}_m := \mathcal{K}_m(A, r_0)$. We seek to minimize the residual on the affine subspace $x_0 + \mathcal{K}_m$ of dimension m.

Definition 2. Choose $v_1 = r_0/\|r_0\|$ for the Arnoldi basis, and let $\beta := \|r_0\|$. Any vector in $x_0 + \mathcal{K}_m$ can be written as $x = x_0 + V_m y$, where $y \in \mathbb{R}^m$. By the above relation, the objective is given as: [4, 6.5.1]

$$\begin{aligned} J(y) &= \|b - A(x_0 + V_m y)\|_2 = \|r_0 - AV_m y\|_2 \\ &= \|\beta v_1 - V_{m+1}\overline{H}_m y\|_2. \end{aligned}$$

By orthonormality of V_{m+1} column vectors, $J(y) = \|\beta e_1 - \overline{H}_m y\|$. The **GMRES approximation** is then the unique vector of $x_0 + \mathcal{K}_m$ which minimizes J.

This results in Algorithm 2 ([4, Algorithm 6.9]).

Algorithm 2 GMRES

Compute $r_0 = b - Ax_0$, $\beta := \|r_0\|_2$, and $v_1 := r_0/\beta$
For $j = 1, 2, \dots, m$ **Do**:
  $\langle$ step n of Arnoldi iteration, Algorithm 1 $\rangle$
EndDo
Define the $(m+1) \times m$ Hessenberg matrix $\overline{H}_m = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m}$
Compute y_m the minimizer of $\|\beta e_1 - \overline{H}_m y\|_2$ and $x_m = x_0 + V_m y_m$

Computing y_m is an $(m+1) \times m$ matrix least squares problem with Hessenberg structure that can be solved via QR factorization in the usual manner, at the cost of $\mathcal{O}(m^2)$ operations. To save further work, rather than construct QR factorizations of the successive matrices $\overline{H}_1, \overline{H}_2, \dots$ independently, we can use an updating process to get the QR factorization of $\overline{H}_m$ from that of $\overline{H}_{m-1}$. Doing so requires a single **Givens rotation** and $\mathcal{O}(m)$ work [5, Mechanics of GMRES].

This approach also enables obtaining the residual norm of the approximate solution without computing x_k, thus allowing to decide when to stop the algorithm inexpensively [1, 3.2]. Saad [1], [4] gives the detailed implementation.

Consider now the algorithm from a practical viewpoint. As m increases, the computational cost increases at least as $\mathcal{O}(m^2 n)$ because of the Gram-Schmidt orthogonalization. The memory cost increases as $\mathcal{O}(mn)$. For large n this limits the largest value of m that can be used. One remedy is to restart the algorithm periodically [4, 6.4.1, 6.5.5]. This is the **restarted GMRES** or GMRES(m) algorithm. Note that GMRES(m) does not necessarily converge, unless the matrix A is positive definite [1, 3.4].

V. Convergence of GMRES

Because $\|r_m\|$ is as small as possible for the subspace $\mathcal{K}_m$, by enlarging $\mathcal{K}_m$ to the space $\mathcal{K}_{m+1}$ we can only decrease the residual norm, or at worst leave it unchanged. Thus we observe that GMRES converges monotonically: $\|r_{m+1}\| \le \|r_m\|$. After n steps (for an $n \times n$ problem), we have $\mathcal{K}_n = \mathbb{C}^n$ and the full GMRES algorithm is guaranteed to converge. However, for GMRES to be useful, it must converge to satisfactory precision in $n \ll m$ steps [5, p.270].

To obtain more useful information about convergence, we must turn to the following problem. The iterate $x_m \in x_0 + \mathcal{K}_m$ can be written as $x_m = x_0 + q(A)r_0$, where q is a polynomial of degree $m-1$.

Furthermore, as x_m minimizes $\|r_m\| = \|b - Ax_m\|$ in the affine subspace $x_0 + \mathcal{K}_m$, we have $x_m = x_0 + q_m(A)r_0$, where q_m is the polynomial that solves: [4, Lemma 6.31]

$$\begin{aligned} \|r_m\| = \|b - A(x_0 + q_m(A)r_0)\| &= \min_{q \in P_{m-1}} \|(I - Aq(A))r_0\| \\ &= \min_{p \in P_m, p(0)=1} \|p(A)r_0\|. \end{aligned}$$

The polynomial p is determined by GMRES. We can show:

Theorem 3 ([4, Proposition 6.32])**.** *Suppose A is diagonalizable, satisfying $A = X\Lambda X^{-1}$ with $\Lambda = diag\{\lambda_1, \lambda_2, \dots, \lambda_n\}$*

is the diagonal matrix of eigenvalues. We have the following upper bound on the residual:

$$\|r_m\| \leq \kappa_2(X) \min_{p\in\mathbb{P}_m, p(0)=1} \max|p(\lambda_i)|.$$

This theorem can be summarized in words as follows. If A is not too far from normal in the sense that $\mathcal{K}(X)$ is not too large, and if properly normalized degree m polynomials can be found whose size on the spectrum $\Lambda(A)$ decreases quickly with m, then GMRES converges quickly [5, p.271].

Example 4. Let $n = 4000$, with entries sampled from $\mathcal{N}(2, 0.5/\sqrt{n})$. The eigenvalues are roughly uniformly distributed in the disk of radius $\frac{1}{2}$ centered at $z = 2$. $\|p(A)\|$ is then approximately minimized by $p(z) = (1 - z/2)^m$, with a convergence rate of 4^{-m} [5].

Corollary 5 ([1])**.** *Suppose A is positive real with symmetric part $M = \frac{1}{2}(A + A^T)$. We have the following upper bound on the residual:*

$$\|r_m\| \leq \left(1 - \frac{\lambda_{\min}(1/2(A^T + A))^2}{\lambda_{\max}(A^T A)}\right)^{m/2} \|r_0\|.$$

In particular, if A is symmetric positive definite, we have:

$$\|r_m\| \leq \left(\frac{\kappa(A)^2 - 1}{\kappa(A)^2}\right)^{m/2} \|r_0\|,$$

where $\kappa = \|A\|\|A^{-1}\|$ denotes the condition number of A.

VI. Preconditioning

For any nonsingular $n \times n$ matrix M, the system

$$M^{-1}Ax = M^{-1}b$$

has the same solution as the nonsingular system $Ax = b$. Solving this system depends on the properties of $M^{-1}A$, and may require fewer steps to converge than the original system [4, 10.1]. We call M the **preconditioner**. The preconditioner can be either applied to the left or right. Latter leads to the system

$$AM^{-1}u = b,$$
$$x \equiv M^{-1}u.$$

This amounts to making the change of variables $u = Mx$, and solving the system with respect to the unknown u [4, 9.1].

M should meet the following requirements:

- M lies between $M = A$ (for the equally hard to solve system $x = A^{-1}b$) and $M = I$ (for the original system $Ax = b$).
- M^{-1} is inexpensive to apply to an arbitrary vector.
- M is "close enough" to A, in the sense that the preconditioned system $M^{-1}Ax = M^{-1}b$ converges more quickly than the system $Ax = b$. Specifically, we want $\|I - M^{-1}A\|$ to be small and the eigenvalues of $M^{-1}A$ close to 1, or clustered around some other value [5, Lecture 40].

Note the following differences when preconditioning GMRES: [4, 9.3]

- Left preconditioning:
  - builds an orthogonal basis of the Krylov subspace
  $$\text{span}\{r_0, M^{-1}Ar_0, \ldots, (M^{-1}A)^{m-1}r_0\};$$
  - obtains the residuals relative to the preconditioned system, i.e. $M^{-1}(b - Ax_m)$;
  - uses one preconditioning operation at the beginning of the outer loop, $r_0 = M^{-1}(b - Ax_0)$.
- Right preconditioning:
  - builds an orthogonal basis of the Krylov subspace
  $$\text{span}\{r_0, AM^{-1}r_0, \ldots, (AM^{-1})^{m-1}r_0\};$$
  - obtains the residuals relative to the initial system, i.e. $b - Ax_m$;
  - uses one preconditioning operation at the end of the outer loop, $x_m = x_0 + M^{-1}V_m y_m$.

A. Polynomial preconditioning

Consider the characteristic polynomial χ_A of A. With the Caley-Hamilton theorem, we have

$$\chi_A(A) = a_n A^n + a_{n-1}A^{n-1} + \cdots + a_1 A + a_0 I = 0,$$

and so, for $a_0 \neq 0$,

$$A^{-1} = \frac{1}{a_0}(-a_n A^{n-1} - a_{n-1}A^{n-2} - \cdots - a_1 I).$$

It thus makes sense to approximate A^{-1} with a polynomial $s(A)$. We set $M^{-1} = s(A)$, and assume s to have degree $d \ll n$. Note that $s(A)A = As(A)$, so left and right preconditioning build the same Krylov subspace. We call M^{-1} a **polynomial preconditioner**.

The matrix $s(A)$ does not need to be formed explicitly: $s(A)v$ can be computed for any vector v from a sequence of matrix-by-vector products [4, 12.3]. In particular, **Horner's method** computes $s(A)v$ with d matrix-vector products:

$$\begin{aligned} s(A)v &= s_{d+1}A^d v + s_d A^{d-1}v + \cdots + s_1 v \\ &= A(A\cdots(s_{d+1}Av + s_d v) + \cdots + s_2 v) + s_1 v. \end{aligned}$$

Remark 6. *A polynomial preconditioner may be combined with other preconditioners M', such as ILU(k), by solving the system:*

$$s(M'^{-1}A)M'^{-1}A = s(M'^{-1}A)M'^{-1}b.$$

We now compute a polynomial preconditioner related to the GMRES polynomial, corresponding to a minimal residual $\|p(A)r_0\| = \|(I - Aq(A))r_0\|$. For a preconditioner, we compute $q(A)$ instead. Denote this polynomial by $M^{-1} = s(A)$ as above, and let $v_0 \neq 0$. We solve:

$$\min\|(I - M^{-1}A)v_0\| = \min_{q\in\mathbb{P}_d}\|v_0 - Aq(A)v_0\|$$

by building a power basis $Y = \{v_0, Av_0, \ldots, A^d v_0\}$, and solving the normal equations

$$(AY)^T AYy = (AY)v_0.$$

The elements of y are the coefficients of M^{-1}. Note the following:

- The method requires no knowledge of the spectrum of A, unlike Chebyshev or least squares polynomials [9].
- The method becomes unstable as the columns of Y lose linear independence, but its results are generally sufficient for low-degree (up to $d = 10$) polynomials [9].
- The coefficients of the polynomial depend on the choice of v_0. Using a randomly chosen vector may be preferable in this case [10].
- The matrix AY is a matrix of dimension $n \times (d+1)$, and takes $d+1$ matrix-vector products to compute. The product $(AY)^T AY$ is of dimension $(d+1) \times (d+1)$ and requires a matrix-matrix multiplication. With $d \ll n$, the system $(AY)^T AYy = (AY)v_0$ is of small dimension and can be solved with a direct method such as LU decomposition.

VII. Numerical experiments

For every problem, the right-hand side is sampled from $\mathcal{N}(0,1)$, the real normal distribution of mean 0 and standard deviation 1. Orthonormalization was done using the Modified Gram-Schmidt procedure. Right preconditioned GMRES is used for comparing residuals $\|b - Ax_i\|$ between the initial and preconditioned systems. The vector v_0 for computing the power basis Y is sampled from $U[-1,1]$, the uniform distribution on the interval $[-1,1]$.

Some of the problems considered are very difficult for GMRES to solve, and, considering the small dimension n, may be better suited for direct solvers. They nonetheless give a good indication of the difficulties involved when using GMRES.

1) The **Circle Eigenvalue Matrix** of size $n = 2000$, a block diagonal matrix with 2×2 blocks:
$$\begin{pmatrix} 1+\cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & 1+\cos(\alpha) \end{pmatrix},$$
$\alpha \in 2k\frac{\pi}{n},\ k \in \{0, 2, \ldots, n-2\}$. All eigenvalues of this matrix are on the unit circle on the complex plane with center $(1,0)$ [9]. The condition number is estimated at 6.3662×10^3.
2) The **Bidiagonal Matrix** of size $n = 5000$ with diagonal elements 0.1, 0.2, 0.3, ..., 0.9, 1, 2, 3, ..., 4990, 4991 (*BiDiag1*). The superdiagonal elements are all set to 0.2 [9]. The symmetric part is positive definite, and the condition number estimated at 8.4589×10^4.
3) As problem 2, but the diagonal elements are set to 10, 11, 12, ..., 5009 (*BiDiag2*). The condition number is estimated at 5.0137×10^3.
4) **S1RMQ4M1** (Matrix Market), a real symmetric positive definite matrix of size $n = 5489$. The matrix has high condition number, estimated at 3.21×10^6 by Matrix Market.
5) **E20R0100** (Matrix Market), a real non-symmetric indefinite matrix of size $n = 4241$. The matrix has high condition number, estimated at 2.15×10^{10} by Matrix Market. This matrix serves as a counter-example for classical preconditioners such as ILU, where their application may result in a worse condition number.
6) **SHERMAN5** (Matrix Market), a real non-symmetric matrix of size $n = 3312$. The condition number is estimated at 3.9×10^5 by Matrix Market. This matrix serves as an example where polynomial preconditioning can worsen convergence.

Polynomials of degree 3, 5, 7, and 10 are used for most problems. Implementation was done in MATLAB[1], based on `gmres.m` from the Netlib Repository.

A. Numerical properties

As explained in Section VI, we expect a preconditioner to be effective if the eigenvalues of $M^{-1}A$ are clustered around a constant value, such as 1. For most test problems, after applying the polynomial preconditioner this was indeed the case. The results are summarized in Table I. The spectrum of E20R0100 after applying a degree 10 polynomial is shown in Fig. 1.

For the symmetric positive definite matrix S1RMQ4M1, preconditioning with degree 7 and degree 9 polynomials lead to negative eigenvalues. These eigenvalues were outliers, and the behavior was not observed for polynomials of a different degree. One may however expect that $M^{-1}A$, while symmetric, is not positive definite in general. This limits the use of the preconditioner with the Preconditioned CG method [2], which depends on a symmetric positive definite preconditioner.

For most test problems, the condition number was inversely proportional to the degree of the applied polynomial. In those cases, a degree 10 polynomials decreased the condition number by a degree of magnitude. The exception was SHERMAN5, where the condition number varied strongly for different degree polyomials. This is illustrated in Fig. 2. We will consider this special case in the following section.

B. Convergence

We now consider convergence of the studied test problems. A faster convergence rate in the preconditioned system leads, for a certain tolerance, to a lower amount of steps in the (restarted) GMRES method. This leads in turn to a smaller number of inner products (see Table II). We also study the impact of polynomial preconditioning on the amount of sparse matrix-vector products.

TABLE I
Extreme values, mean μ, and standard deviation σ for $|\lambda(A)|$, the (magnitude of the) spectrum of A.

Matrix A	deg	$\min(\lvert\lambda(A)\rvert)$	$\max(\lvert\lambda(A)\rvert)$	$\mu(\lvert\lambda(A)\rvert)$	$\sigma(\lvert\lambda(A)\rvert)$
CEM	–	0.00314	2	1.2742	0.61523
CEM	10	0.01727	1.2070	0.94540	0.14954
BiDiag1	–	0.1	4991	2491.5	1443.5
BiDiag1	3	0.00024	1.1509	0.95895	0.19989
S1RMQ4M1	–	0.37970	6.8743e+05	8.6760e+04	1.2595e+05
S1RMQ4M1	3	7.7127e-06	1.2028	0.47948	0.49665
E20R0100	–	1.3999e-06	13.057	4.3537	4.0740
E20R0100	10	4.8582e-06	1.0902	0.71268	0.44947

[1]See https://github.com/fvanmaele/polynomial-preconditioning for the implementation and supplementary materials.

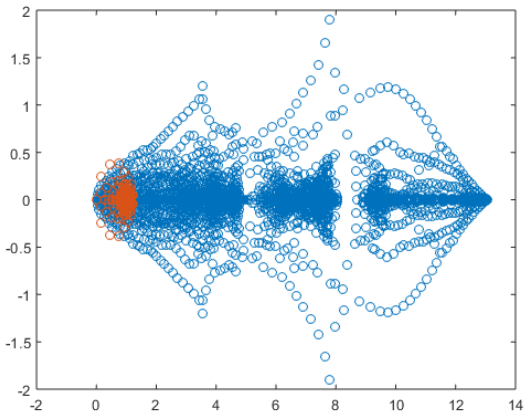

Fig. 1. Spectrum of A (blue) and $M^{-1}A$ (orange, $d = 10$) for E20R0100

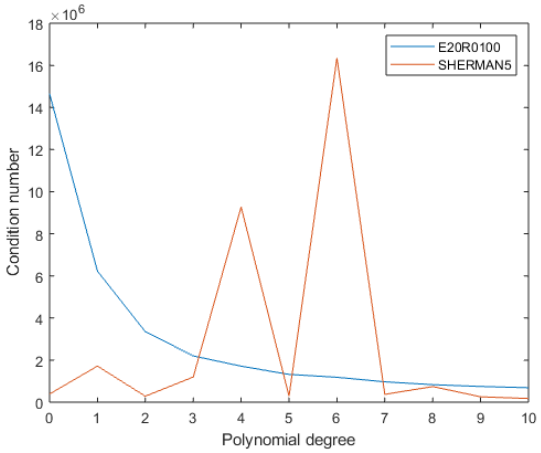

Fig. 2. Condition numbers for E20R0100 (blue) and SHERMAN5 (orange), depending on the polynomial degree of the preconditioner.

In the below, the maximum of iterations is set to 20,000, and the tolerance to $\varepsilon = 10^{-8}$ for the relative residual $\frac{\|b-Ax_m\|}{\|b\|}$.

- **BiDiag1** and **BiDiag2** converged without preconditioning using GMRES(20), reaching $\varepsilon = 10^{-8}$ in 18,193 and 258 steps, respectively. Preconditioning increased the convergence rate, reaching the tolerance in 1,786 and 60 steps, respectively, with a degree 3 polynomial.
- **S1RMQ4M1** did not converge with GMRES(50) below 20,000 steps, and a polynomial of degree 5 was required to converge (in 11,901 steps). Higher degree polynomials lead to better results. See Fig. 3.
- The **Circle Eigenvalue Matrix** did not converge even with GMRES(100), but a degree 3 polynomial sufficed for convergence in 12,564 steps.
- **E20R0100** is a more difficult problem, and required a degree 7 polynomial to converge (in 9,500 steps) with GMRES(100). The problem converged in only 997 steps if we raised the polynomial degree to 10. See Fig. 5.
- Results for **SHERMAN5** varied between samples of v_0 of b. Loe [10] suggested degree 7 polynomials for improved convergence. For the tested samples (seed $\in \{42, 666, 1892\}$), we had more consistent results with a degree 3 polynomial. In this case, eigenvalues of $M^{-1}A$ were mapped to the right half of the complex plane. Other degrees typically resulted in worse convergence. GMRES without preconditioning, however, converged reliably for each sample. See Fig. 4 and 6.

The amount of SpMVs increased by ~3x (for S1RMQ4M1) up to ~6x (for the Circle Eigenvalue Matrix) for preconditioned matrices. For E20R0100, there was up to ~8x increase, but with fast convergence of the degree 10 preconditioner, latter only resulted in an increase of ~1.25x in SpMVs. See Table III for an overview.

Remark 7. *The polynomial preconditioner had very small coefficients for its higher-order terms, for all problems except the Circle Eigenvalue Matrix. This effect may originate from the ill-conditioned nature of computing coefficients with the normal equations and a power basis [10, III]. In particular, LAPACK reported a very small* RCOND *for these problems (below* 10^{-17} *for E20R011 and below* 10^{-25} *for S1RMQ4M1).*

CONCLUSION

Polynomial preconditioners can be used along with standard preconditioning, and are effective to implement on parallel systems as a series of sparse matrix-vector products. We have introduced a polynomial preconditioner related to the minimum residual (GMRES) polynomial.

To compute the coefficient of the polynomial, no knowledge of the spectrum of A is required. It suffices to take a random vector v_0, and solve the least squares problem $\min \|v_0 - s(A)Av_0\|$. This however bases on computing a power basis $(A^i v_0)$, which is ill-conditioned. Therefore the degree of the polynomial is limited, typically to $d \leq 10$.

TABLE II
INNER PRODUCTS FOR POLYNOMIALS OF DIFFERENT DEGREE

Inner p. \ deg	**deg = 0**	**deg = 3**	**deg = 5**	**deg = 7**	**deg = 10**
CEM	1,010,000	633,330	304,176	183,285	92,385
BiDiag1	190,981	18,711	9,261	5,526	2,541
BiDiag2	2,691	630	423	288	153
S1RMQ4M1	510,000	510,000	303,451	171,553	84,051
E20R0100	1,010,000	1,010,000	1,010,000	479,750	50,203

TABLE III
SPMVS FOR POLYNOMIALS OF DIFFERENT DEGREE

SpMVs \ deg	**deg = 0**	**deg = 3**	**deg = 5**	**deg = 7**	**deg = 10**
CEM	20,401	50,886	36,715	29,565	20,622
BiDiag1	20,013	7,594	5,631	4,491	2,862
BiDiag2	284	255	273	274	209
S1RMQ4M1	20,801	82,001	73,079	55,111	37,070
E20R0100	20,401	81,001	121,401	76,855	11,087

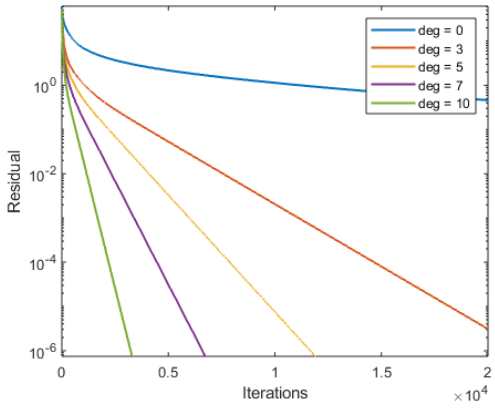

Fig. 3. Convergence of S1RMQ4M1, GMRES(50)

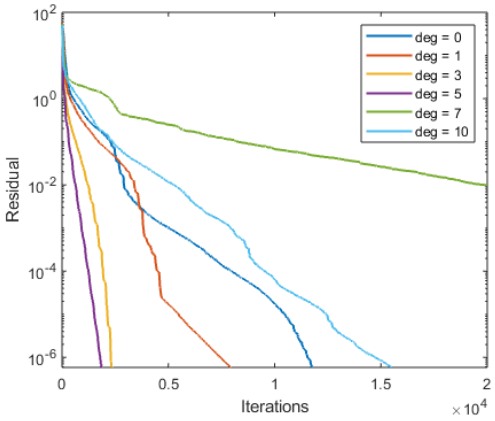

Fig. 4. Convergence of SHERMAN5, GMRES(100), seed = 42

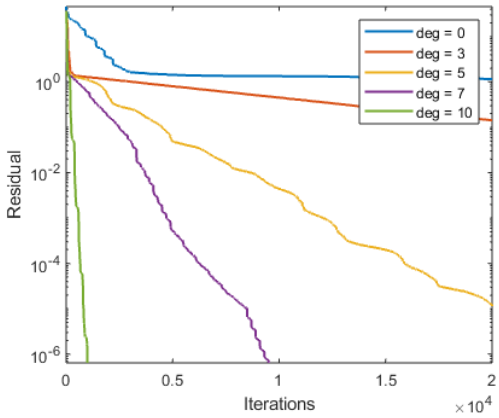

Fig. 5. Convergence of E20R0100, GMRES(100)

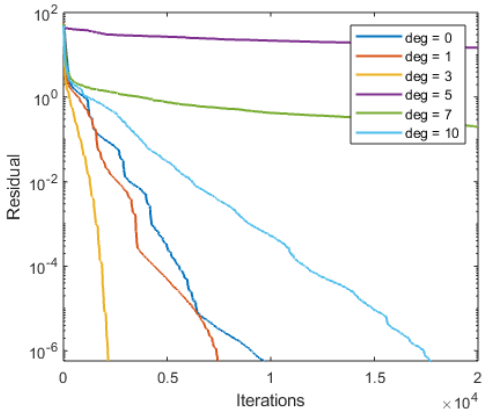

Fig. 6. Convergence of SHERMAN5, GMRES(100), seed = 1892

Applying this preconditioner to several difficult problems, including E20R0100, shows it can be used to significantly improve convergence of GMRES. In particular, on parallel systems, synchronization points from inner products in each GMRES step are reduced.

Care should be taken for problems such as SHERMAN5, where effectiveness varies for different samples of the vector v_0, and can even result in worse convergence.

REFERENCES

[1] Y. SAAD, M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. Stat. Comput. Vol. 7, No. 3, July 1986.

[2] Y. SAAD, *Practical use of polynomial preconditionings for the conjugate gradient method*, SIAM J. Sci. Stat. Comput. Vol. 6, No. 4, October 1985.

[3] F. MAGOULES, F. ROUX, G. HOUZEAUX, *Parallel Scientific Computing,* Wiley, 2015.

[4] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.

[5] L. N. TREFETHEN, D. BAU, *Numerical Linear Algebra*, SIAM, 1997.

[6] M. BENZI, G. H. GOLUB, J. LIESEN, *Numerical solution of saddle point problems,* Acta Numerica (2005), pp. 1-137.

[7] M. ROZLOŽNÍK, *Saddle-Point Problems and Their Iterative Solution*, Birkhäuser, 2018.

[8] R. BARRET et. al., *Templates for the Solution of Linear Sytems: Building Blocks for Iterative Methods*, SIAM, 1994.

[9] Q. LIU, R. B. MORGAN, W. WILCOX, *Polynomial Preconditioned GMRES and GMRES-DR*, SIAM J. Sci. Comput. Vol. 37, No. 5, pp. S407-S428, 2015.

[10] J. A. LOE, H. K. THORNQUIST, E. G. BOMAN, *Polynomial Preconditioned GMRES to Reduce Communication in Parallel Computing*, arXiv, Oct 2019.

[11] V. HERNANDÉZ, J. E. ROMÁN, A. TOMAS, *A Parallel Variant of the Gram-Schmidt Process with Reorthogonalization*, Proceedings of the International Conference Parco 2005, 13-16 September 2005.