Krylov Methods and Polynomials

Ron Morgan

Abstract

It is well-known that all Krylov methods have polynomials underlying them. Here these polynomials are used in new ways to develop iterative methods. There are several applications including systems of linear equations with multiple right-hand sides.

1 Polynomial Preconditioning

Polynomial preconditioning goes way back, see for instance [6, 11, 10, 2, 12], but has never become standard. In [5, 8], polynomial preconditioning is made a more practical method with these improvements:

1. The polynomial is easy to determine. It is generated by GMRES instead of from eigenvalue estimates as some approaches have done.
2. The implementation is efficient due to using roots of the GMRES residual polynomial.
3. The method is more stable than some other approaches. This is from the implementation with roots instead of coefficients. Also, additional stability control comes from adding extra copies of outstanding roots.

With polynomial preconditioning, $Ax = b$ becomes

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

Polynomial preconditioning works because it transforms the spectrum of A into a better spectrum. Another reason why it is effective is that there is more power per iteration and much less orthogonalization. Also, there is greater opportunity for parallelism. However, polynomial preconditioning may not needed for fairly easy systems.

Example Polynomial preconditioning is used for a matrix from a biharmonic differential equation. A degree 200 polynomial is applied, then a degree 50 polynomial is used along with an incomplete factorization preconditioner. Table 1 first shows that polynomial preconditioning can be very effective for the difficult original system of equations. The time is reduced from hours to under a minute. The problem is easier when standard incomplete factorization preconditioning is applied. Then the effect of the polynomial preconditioning is not as dramatic, but it is still helpful. The time goes from over 3 minutes to below 9 seconds. See [8] for more results.

Table 1: Polynomial preconditioning applied to a biharmonic matrix of size n = 40,000 from the differential equation $-u_{xxxx} - u_{yyyy} + u_{xxx} = f$ on the unit square. IF stands for incomplete factorization with no fill-in after shifting the matrix by $0.5 * I$.

Method	GMRES(50)	PP(200)-GMRES(50)	IF-GMRES(50)	PP(50)-IF-GMRES(50)
Time	14.6 hours	55 seconds	3.5 minutes	8.5 seconds

2 Polynomial Approximation of A^{-1}

It is surprising that even for a large matrix, it is often possible to find a polynomial p so that $p(A)$ is a good approximation to A^{-1} [4]. We use the p polynomial from Equation (1) that is generated by GMRES. This is implemented using the roots of the GMRES residual polynomial.

It can be proved for the symmetric case that the accuracy of the approximating polynomial follows the residual norm curve. Stability control with added roots is even more important here than for polynomial preconditioning, because a high degree polynomial is needed.

Example: For a convection-diffusion matrix, Table 2 shows that the relative accuracy of the polynomial approximation to the inverse follows the GMRES residual norm. See [4] for more.

Table 2: A polynomial approximation is found to the inverse of a matrix of size $n = 2500$ from the convection-diffusion equation $-u_{xx} - u_{yy} + 2u_x = f$ on the unit square. The starting vector is a random vector normed to one. Relative accuracy of the polynomial is compared to the GMRES residual norm.

Degree	GMRES residual norm	$\frac{\|A^{-1}-p(A)\|}{\|A^{-1}\|}$
50	$8 * 10^{-3}$	$2 * 10^{-1}$
100	$4 * 10^{-5}$	$6 * 10^{-4}$
150	$1 * 10^{-8}$	$3 * 10^{-7}$
200	$8 * 10^{-12}$	$3 * 10^{-10}$

Applications of polynomial approximation to A^{-1} include:

1. Systems with multiple right-hand sides [4]. The polynomial approximation that is generated with one right-hand side can be applied to other right-hand sides to solve the systems.
2. Multilevel Monte Carlo for the trace of the inverse in quantum chromodynamics [7]. Polynomials form the basis for this approach, but deflation of eigenvalues is also crucial.
3. Functions of matrices (in progress). A polynomial approximation to the function of a matrix can be found by interpolating the function at the harmonic Ritz values (the roots of the GMRES polynomial).

3 Twin CG for multiple right-hand side systems. Also Twin CR, BiCGStab, BiCG, GMRES and QMR.

The first application in the previous section used the same polynomial for multiple right-hand sides. Here we again have the same polynomial, but applied in a simpler way. We use the same iteration coefficients for all right-hand sides. When this approach is used for the conjugate gradient method (CG), we call this "Twin CG for multiple right-hand sides."

As an example, consider two systems, $Ax^{(1)} = b^{(1)}$ and $Ax^{(2)} = b^{(2)}$. One step in the CG iteration for the first system is $x^{(1)} = x^{(1)} + \alpha^{(1)} * p^{(1)}$. Our approach is to also do $x^{(2)} = x^{(2)} + \alpha^{(1)} * p^{(2)}$ with the same $\alpha^{(1)}$ as for the first system. Similarly, the other CG steps have the same constants for both right-hand sides.

The residual polynomial helps explain why this is effective. For the second right-hand side, the residual polynomial is the same as for the first. So if $r^{(1)} = \pi^{(1)}(A)b^{(1)}$, then also $r^{(2)} = \pi^{(1)}(A)b^{(2)}$.

The polynomial $\pi^{(1)}$ needs to be small at the eigenvalues of A in order for the first system to converge, and this makes the second system converge also. The first right-hand side, $b^{(1)}$, does need to not be deficient in some eigencomponents. For the deficient case, creating a first system with a random right-hand side is recommended.

With this twin approach for CG, the multiple right-hand systems generally converge together. However, there can be momentary instability due to steep slope of the polynomial at an outstanding eigenvalue. This tends to be quickly automatically corrected.

Remarkable things about Twin CG:

1. The code can be very simple. All x vectors for all right-hand sides can be grouped together into one matrix, and similarly for r and p vectors. Then most operations can be done with matrices instead of vectors.
2. This method is extremely parallelizable. All dot products are eliminated except for the first right-hand side (one may need to monitor residual norms for other right-hand sides occasionally after the first system converges). The matrix-vector products can be done together for all right-hand sides. The DAXPY's can also be performed together, so they become matrix operations.
3. Stability control happens automatically due to roundoff error. *So roundoff error is essential to the success of the method.* This is because the Lanczos algorithm that is the basis for CG develops extra copies of outstanding eigenvalues [9]. This mimics the addition of roots given in [5, 8] for stability control with polynomial preconditioning. With this Twin CG approach, it is surprising that the extra copy appears almost as soon as stability control is needed. However, for cases with several outstanding eigenvalues, extra copies will not all appear simultaneously, so we have a way of augmenting with some manual stability control.
4. Seed methods [3, 1] for deflating eigenvalues can be added. Seeding is done during solution of the first system, then the Twin CG method is applied to the second and subsequent systems. In addition, we are working on a new seed approach for the case where roundoff error interferes with the accuracy of the seeding [1].

Finally, a few comments about nonsymmetric systems:

1. Similar to CG for symmetric systems, a Twin BiCG method can be given for the nonsymmetric case. BiCG has automatic stability control similar to CG.
2. Twin BiCG uses only half of the matrix-vector products for the second and subsequent right-hand sides as for the first.
3. BiCGStab does not have the automatic stability control.
4. The twin approach for multiple right-hand sides can even be used with restarted GMRES. Dot products are eliminated except for the first right-hand side, but the rest of the orthogonalization cost is needed for each right-hand side system.

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] T. F. Chan and W. Wan, *Analysis of projection methods for solving linear systems with multiple right-hand sides*, SIAM J. Sci. Comput., 18 (1997), pp. 1698–1721.

[4] M. Embree, J. A. Henningsen, J. Jackson, and R. B. Morgan, *Polynomial approximation to the inverse of a large matrix*, tech. rep., Baylor University, 2024. Available at https://sites.baylor.edu/ronald_morgan/reports/.

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

[6] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 33–53.

[7] P. Lashomb, R. B. Morgan, T. Whyte, and W. Wilcox, *Multi-polynomial Monte Carlo for QCD trace estimation*, Comput. Phys. Commun., 300 (2024), pp. 109163–1 – 109163–10.

[8] J. A. Loe and R. B. Morgan, *Toward efficient polynomial preconditioning for GMRES*, Numer. Linear Algebra Appl., 29 (2021), pp. 1–21.

[9] C. C. Paige, *The computation of eigenvectors and eigenvalues of very large sparse matrices*. PhD Thesis, University of London, 1971.

[10] Y. Saad, *Practical use of some Krylov subspace methods for solving indefinite and unsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 5 (1984), pp. 203–228.

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

[12] 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.