Polynomial Preconditioning for RRGMRES

Jennifer Loe, Ron Morgan

Baylor University

23 May 2018

Preconditioning with the GMRES Polynomial:

Used for

- GMRES [Liu, Morgan, Wilcox]
- BiCGStab and IDR [L., Morgan]
- Arnoldi for Eigenvalues [Embree, L., Morgan]
- Communication Avoiding with GMRES [Boman, L., Thornquist]

Preconditioning with the GMRES Polynomial:

Used for

- GMRES [Liu, Morgan, Wilcox]
- BiCGStab and IDR [L., Morgan]
- Arnoldi for Eigenvalues [Embree, L., Morgan]
- Communication Avoiding with GMRES [Boman, L., Thornquist]

Can we use it for ill-posed problems?

Using GMRES Algorithms:

GMRES: [Saad, Schultz]

- Solves large, sparse linear systems $Ax = b$
- For nonsymmetric systems
- Works better if small eigenvalues of A are well-spaced and others are clustered near 1.
- Can solve preconditioned system $AMy = b$, $x = My$

GMRES Algorithm

Big idea: Build a basis for Krylov subspace

$$\mathcal{K}(A, b) = span\{b, Ab, A^2b, \ldots, A^{m-1}b\}.$$

Pick a good approximate solution from $\mathcal{K}(A, b)$.

GMRES Algorithm

Big idea: Build a basis for Krylov subspace

$$\mathcal{K}(A, b) = span\{b, Ab, A^2b, \ldots, A^{m-1}b\}.$$

Pick a good approximate solution from $\mathcal{K}(A, b)$.

1. Multiply A (or AM) times previous basis vector.
2. Orthogonalize against the previous basis vectors. (Gram-Schmidt)
3. Solve a small Hessenberg system.
4. Restart when subspace gets too large.

GMRES Algorithm

Big idea: Build a basis for Krylov subspace

$$\mathcal{K}(A, b) = span\{b, Ab, A^2b, \ldots, A^{m-1}b\}.$$

Pick a good approximate solution from $\mathcal{K}(A, b)$.

1. Multiply A (or AM) times previous basis vector.
2. Orthogonalize against the previous basis vectors. (Gram-Schmidt)
3. Solve a small Hessenberg system.
4. Restart when subspace gets too large.

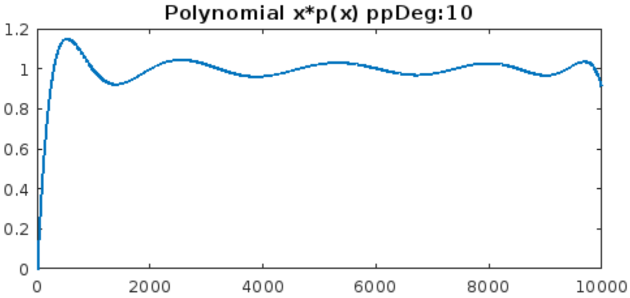
Orthogonalization:

Figure: More vectors $\Longrightarrow$ more work to keep them in line...

Polynomial Preconditioning with The GMRES Polynomial

Solve $Ap(A)p(A)^{-1}x = b$ where $p(A)$ is the GMRES polynomial.

- Separates small eigenvalue of A and clusters others near 1.
- More power for each step of orthogonalization!
- No need to solve linear system to apply preconditioner.
- No eigenvalue estimates needed.



Computing the Polynomial

[Liu, Morgan, Wilcox]

1. Build a power basis $V = [b, Ab, \ldots, A^m b]$.
2. Solve the normal equations

$$(AV)^* AVy = (AV)^* b.$$

3. The elements of y are the coefficients of $p(A)$.

$$p(A) = y_{m+1}A^m + y_m A^{m-1} + \cdots + y_2 A + y_1$$

Improving Convergence for Restarted GMRES

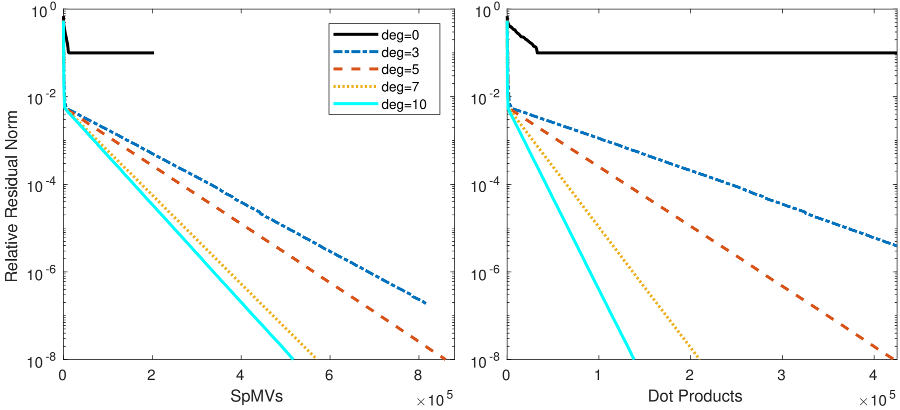


Figure: Residual norm convergence for the matrix e20r0100 with a random right-hand side. Subspace size = 50. Degree 0 indicates no preconditioning. All tests were run to 200000 max iterations.

GMRES Algorithm

Big idea: Build a basis for Krylov subspace

$$\mathcal{K}(A, b) = span\{b, Ab, A^2b, \ldots, A^{m-1}b\}.$$

Pick a good approximate solution from $\mathcal{K}(A, b)$.

1. Multiply A (or AM) times previous basis vector.
2. Orthogonalize against the previous basis vectors. (Gram-Schmidt)
3. Solve a small Hessenberg system.
4. Restart when subspace gets too large.

For RRGMRES: Use subspace $\mathcal{K}(A, Ab) = span\{Ab, A^2b, \ldots, A^mb\}$.

GMRES Algorithm

Big idea: Build a basis for Krylov subspace

$$\mathcal{K}(A, b) = span\{b, Ab, A^2b, \ldots, A^{m-1}b\}.$$

Pick a good approximate solution from $\mathcal{K}(A, b)$.

1. Multiply A (or AM) times previous basis vector.
2. Orthogonalize against the previous basis vectors. (Gram-Schmidt)
3. Solve a small Hessenberg system.
4. Restart when subspace gets too large.

For RRGMRES: Use subspace $\mathcal{K}(A, Ab) = span\{Ab, A^2b, \ldots, A^m b\}$.

Using GMRES Algorithms:

GMRES: [Saad, Schultz]

- Solves large, sparse linear systems $Ax = b$
- For nonsymmetric systems
- Works better if small eigenvalues of A are well-spaced and others are clustered near 1.
- Can solve preconditioned system $AMy = b$, $x = My$

RRGMRES: [Calvetti, Lewis, Reichel]

- Better for noisy right-hand side
- Want to cluster large eigenvalues near 1 and ignore small eigenvalues.

A Simple RRGMRES Example (no restarting):

$n = 10000$
$A = \text{diag}(0, .1, .2, \ldots, 1.9, 2, 3, 4, \ldots, 9981)$
Put $A(500, 500) = 0$ and $A(5000, 5000) = 0$.
$b = [1, 1, \ldots, 1]^T$
rtol ≈ 1.731

Poly Deg	**Iters**	**MVPs**	**Time (s)**
0	1252	1252	143.6
5	324	2926	6.59
10	171	3269	1.87
15	116	3394	1.08
20	88	3472	0.65
25	71	3529	0.48

A Deblurring Problem:

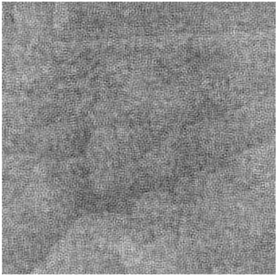
(a) Original Image

(b) Blurred with Noise

With Poly Preconditioning:

Using a polynomial preconditioner of Degree = 10:

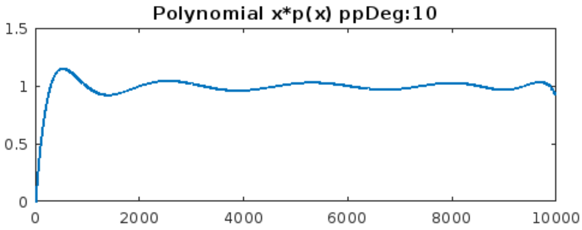
(a) Iteration 2 (30 mvps)



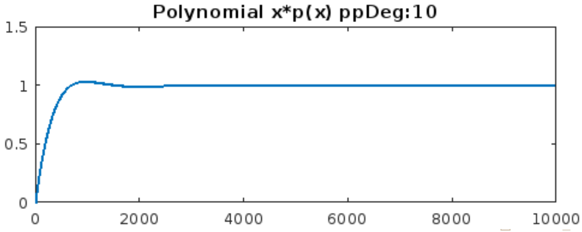
(b) Iteration 4 (50 mvps)

Another Polynomial:

GMRES Polynomial:



RRGMRES Polynomial:



Poly Preconditioning Try 2:

Using an RRGMRES polynomial preconditioner of Degree = 10:

(a) Iteration 2 (31 mvps)

(b) Iteration 4 (51 mvps)

Does it really need a polynomial?

RRGMRES with no preconditioner:

(a) Iteration 1

(b) Iteration 51 (51 mvps)

A final comparison:

(a) Blurred and Noisy

(b) RRGMRES Only

(c) GMRES Poly Prec

(d) RRGMRES Poly Prec

Future Challenges:

Can we use polynomial preconditioning to reduce expenses for RRGMRES and/or image deblurring? (Perhaps combined with other regularization? Different problems? Lower degree polynomial?)

To Be Determined...

The daydreams of cat herders