UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Polynomial Preconditioners

Ferdinand Vanmaele

Introduction

- Discretization of **partial differential equations**
  - Solve $Ax = b$ with A matrix of high dimension n
- Common in science and industry
  - Fluid dynamics
  - Astrophysics
  - Biochemistry
  - Economics
  - etc...

Example: Model problem

- Problem
  - $-u''(x) = f(x)$ for $x \in (0,1)$
    $u(0) = u(1) = 0$
- Discretization
  - $x_i = i \times h$, $i = 0, \dots, n+1$, where $h = 1/(n+1)$
- Central difference approximation
  - $-u_{i-1} + 2u_i - u_{i+1} = h^2 f(x_i)$
- Linear system
  - $Ax = f$ with $A \in R^{n \times n}$

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

Introduction

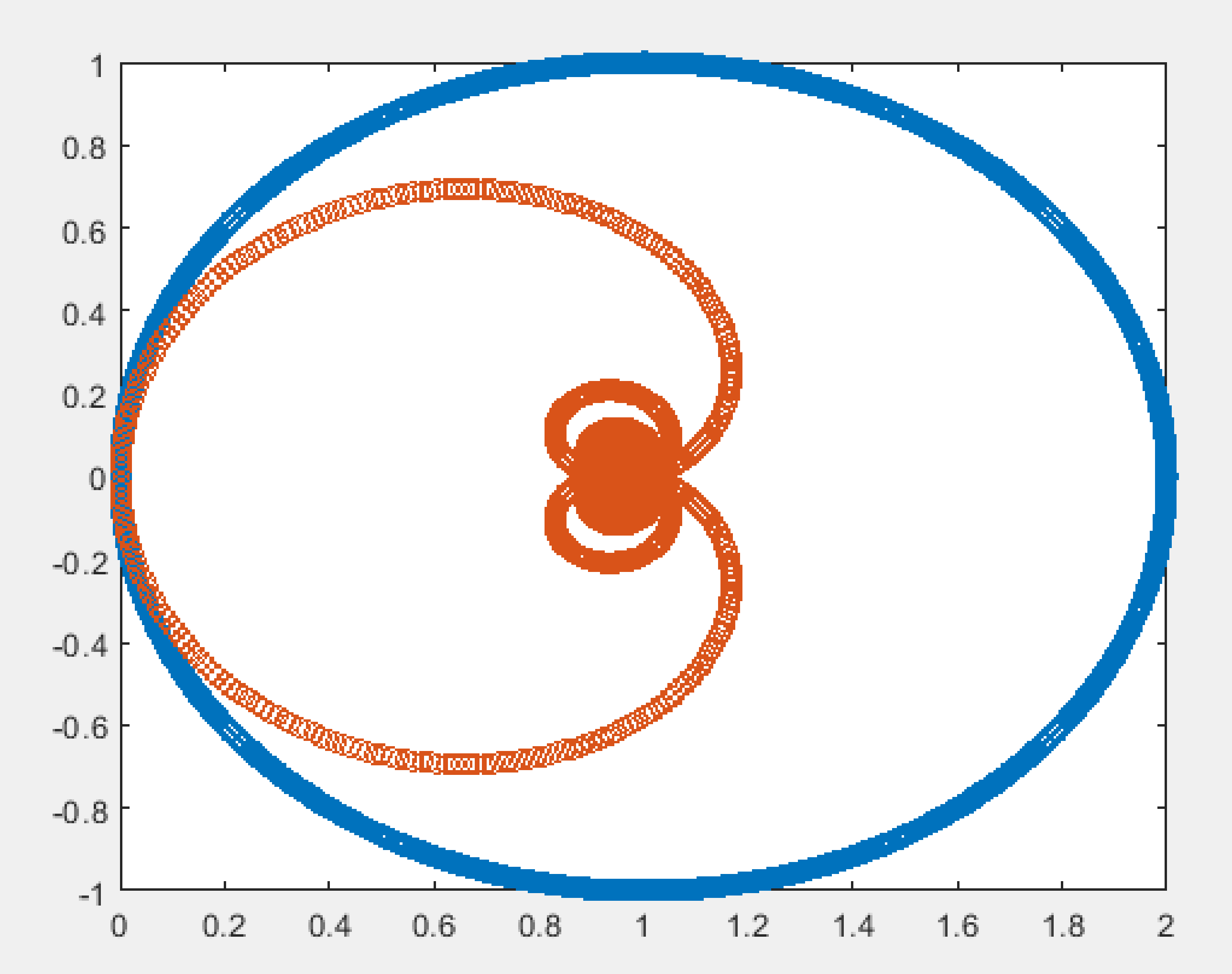
- Matrix A
  - Possible non-symmetric or indefinite
  - Sparse!
- Iterative solvers
  - **Goal:** Reduce storage and complexity requirements
  - Series $(x_i) \to x$ of approximate solutions to $Ax = b$
- Preconditioned system $M^{-1}Ax = M^{-1}b$
  - **Goal:** Improve convergence rate of (x_i)
  - M^{-1} as low-degree polynomial
  - Same solution as $Ax = b$

Example: Circle Eigenvalue Matrix

- A block diagonal, size $n = 2000$
- 2×2 blocks $\begin{bmatrix} 1+\cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & 1+\cos(\alpha) \end{bmatrix}$, $\alpha \in \left\{0, \frac{2\pi}{n}, \frac{4\pi}{n}, \dots, \frac{1998\pi}{n}\right\}$
- Eigenvalues on unit circle in complex plane
- Condition number[1] $K(A) \approx 637$
- Difficult problem without preconditioning

1: $K(A) = \|A^{-1}\|\|A\|$ denotes how sensitive the solution x is to perturbations in the matrix A or the right-hand side b.

Example: Circle Eigenvalue Matrix



- Polynomial $M^{-1} = s(A)$
- $\deg(s) = 10$
- $K(M^{-1}A) \approx 69.9$
- Eigenvalues of M^{-1} in $\mathbb{C}$

Iterative solvers

- Most entries of A are zero
  - Store A as a *sparse matrix*
  - Multiple formats available, e.g. CSR
- Direct methods
  - Decomposition $A = LU$
  - L and U may become dense: *fill-in*
  - High storage and complexity requirements for large n
- Alternative: Iterative methods

Iterative solvers

- Starting approximation x_0
- Method $x_i \to x_{i+1}$
  - $x_{i+1} = x_i + \alpha N^{-1} r_i$ where $r_i = Ax_i - b$ (stationary)
  - $x_{i+1} = x_i + constant_i * search\ direction_i$ (non-stationary)
- Stopping criterion
  - Relative residual: $\frac{\|Ax_i - b\|}{\|b\|} \leq \epsilon$
    - ⚠ $\|Ax_i - b\| \leq \|A\|\|x_i - x\|$
  - Typically $\epsilon = 10^{-1} \ldots 10^{-12}$
- Optimality condition?

Iterative solvers

- **Idea:** Let $N = I$, $\alpha = 1$, $x_0 = 0$. Write out $x_i \to x_{i+1}$:
  - $x_1 = x_0 + r_0 = r_0$
  - $x_2 = x_1 + r_1 = r_0 + (b - Ax_0)$
    $= 2r_0 - Ar_0$
  - $x_3 = x_2 + r_2 = \cdots$
    $= 3r_0 - 3Ar_0 + A^2 r_0$
  - …
- Implication:
  - $x_i \in x_0 + span\{r_0, Ar_0, \dots, A^{i-1} r_0\}$ ⟵ Krylov subspace $\mathcal{K}^i$

Iterative solvers

- **GMRES**
  - Construct orthonormal basis $(v_1 \cdots v_i)$ of $\mathcal{K}_i$
  - Minimize $\|b - Ax_i\|$ for $x_i \in x_0 + \mathcal{K}_i$
  - Works for general matrices A
- Step amount
  - After n steps, $\mathcal{K}_n = \mathbb{R}^n$ and the solution is exact
  - We want $i \ll n$ in practice
- Limitations
  - No short recurrence for general matrices 😥
  - After m steps, set $x_m = x_0$ and restart algorithm: *GMRES(m)*

Iterative solvers

- **Building blocks**
  - Vector updates (SAXPY)
    - Easy to parallelize ✔
  - Inner products
    - Parallelization: synchronization between all processes 🙄
  - Sparse matrix-vector products (SpMV)
    - Parallelization: synchronization between neighbours 😋
  - Matrix-matrix product
    - Determine preconditioner (before iteration start)

Preconditioning

- Solve $M^{-1}Ax = M^{-1}b$
  - Same solution as original system $Ax = b$
- Extreme cases
  - $M = A$: equally hard to solve problem $x = A^{-1}b$
  - $M = I$: original system $Ax = b$
- M "in between" A and I:
  - Matrix norm $\|I - M^{-1}A\|$ small
  - Eigenvalues of $M^{-1}A$ close to 1
- **Goal:** M^{-1} as a polynomial
  - Parallelization properties (SpMVs)

Preconditioning

- M^{-1} matrix polynomial of degree d
  - $M^{-1} = y_{d+1}A^d + y_d A^{d-1} + \cdots + y_2 A + y_1 I$
  - Coefficients $y = (y_1 \cdots y_{d+1})$ determined by A and some $v_0 \neq 0$
- Minimize $\|(I - M^{-1}A)v_0\|$
  - Choose v_0
    - A random vector, e.g. $v_0 \sim \mathcal{N}$(-1,1) gives good results in practice
  - Construct power basis $\mathrm{Y} = \{v_0, Av_0, \dots, A^d v_0\}$
    - ⚠ Columns of Y lose linear independence!
  - Solve least squares problem $\min\|v_0 - s(A)Av_0\|$
    - Solve normal equations $(AY)^T AYy = (AY)v_0$

Preconditioning

- Practical considerations
  - Do not store $\mathrm{M}^{-1} = s(A)$ explicitly (vector products)
    - $z^{(1)} = y_{d+1}Av + y_d v$
    - $z^{(2)} = A * z^{(1)} + y_{d-1}v$
    - $z^{(3)} = A * z^{(2)} + y_{d-2}v$
    - $z^{(d)} = A * z^{(d-1)} + y_1 v$

$\mathrm{M}^{-1}\mathrm{v} = A(A \cdots (y_{d+1}Av + y_d v) + \cdots y_2 v) + y_1 v$

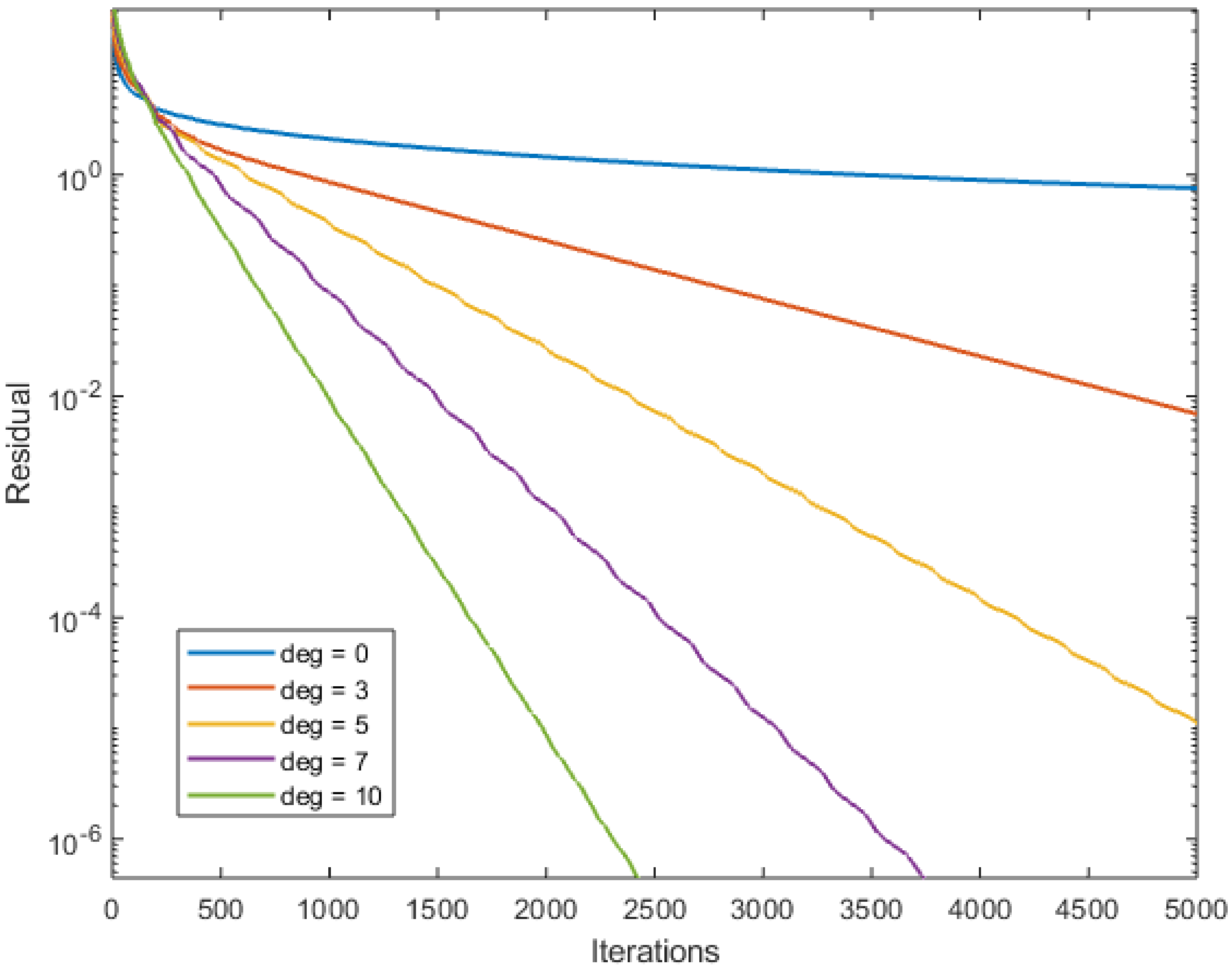
  - Set degree of the polynomial
    - Cost of multiplication $(AY)^T AY$ → lower degree
    - Difficulty of problem $Ax = b$ → higher degree
  - Can be combined with other preconditioners, e.g. ILU(k)

Example: Model problem (n = 5)

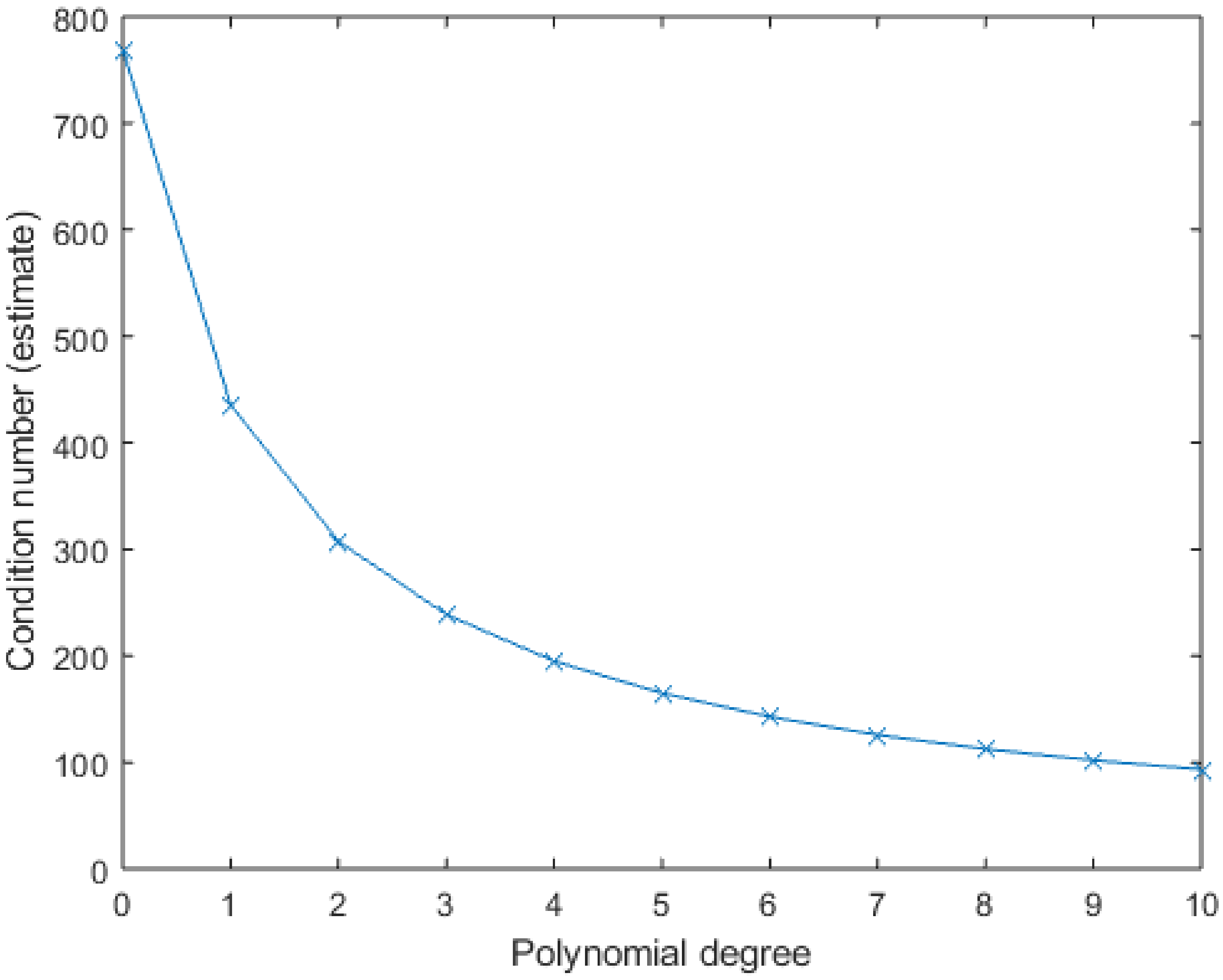
- $A = \frac{1}{36}\begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}$ for $n = 5$. Choose “random” $v_0 = \begin{pmatrix} 1/3 \\ -1 \\ 0 \\ -1 \\ -1 \end{pmatrix}$ and set $d = 2$.

- $Y = (v_0\ Av_0\ A^2v_0) = \begin{pmatrix} \frac{1}{3} & \frac{5}{108} & \frac{17}{3888} \\ -1 & -\frac{7}{108} & -\frac{25}{3888} \\ 0 & \frac{1}{18} & \frac{11}{1944} \\ -1 & -\frac{1}{36} & -\frac{1}{432} \\ -1 & -\frac{1}{36} & -\frac{1}{1296} \end{pmatrix}$, $AY = \begin{pmatrix} \frac{5}{108} & \frac{17}{3888} & \frac{59}{139968} \\ -\frac{7}{108} & -\frac{25}{3888} & -\frac{43}{67625} \\ \frac{1}{18} & \frac{11}{1944} & \frac{13}{23328} \\ -\frac{1}{36} & -\frac{1}{432} & -\frac{12}{45395} \\ -\frac{1}{36} & -\frac{1}{1296} & \frac{1}{46656} \end{pmatrix}$

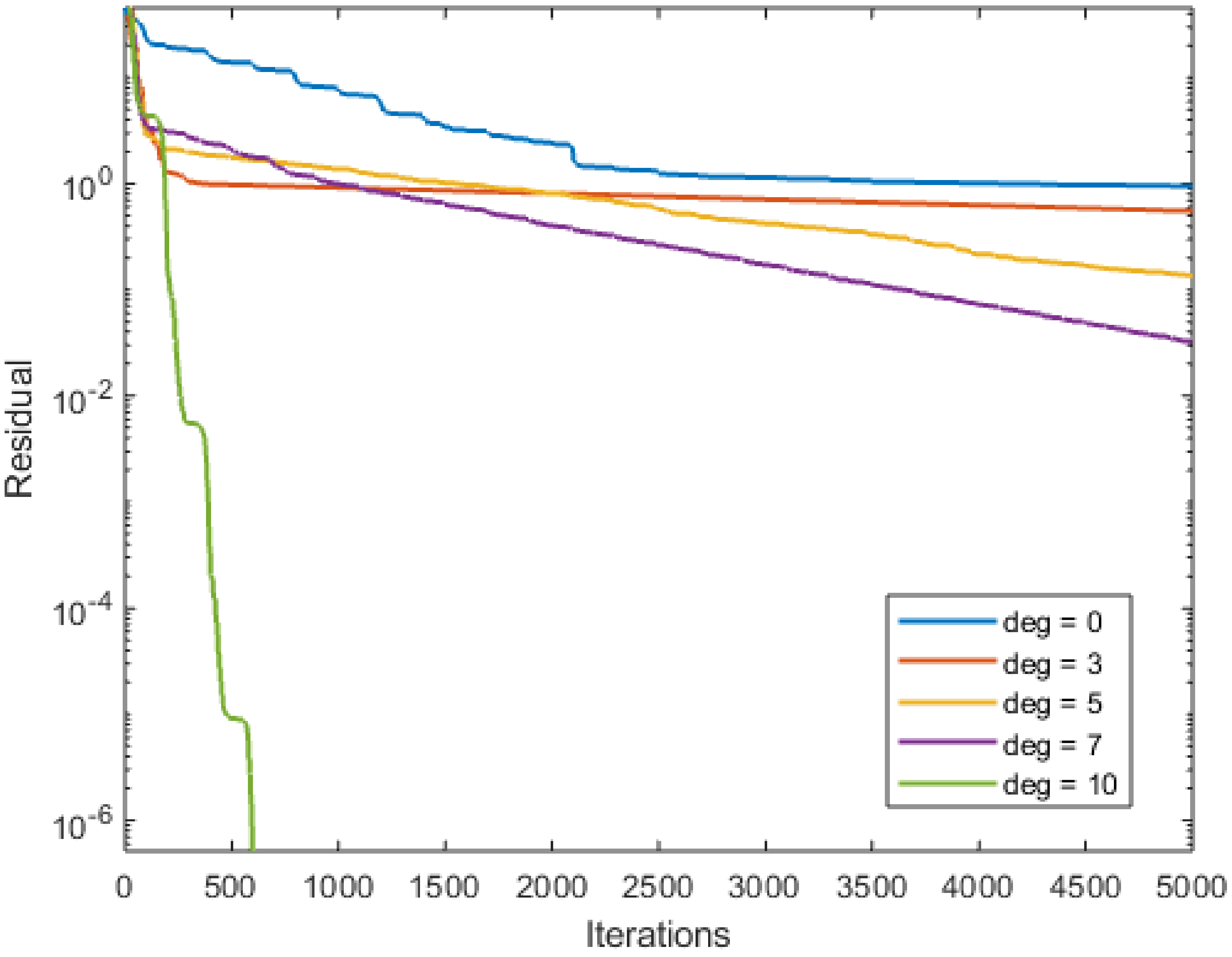
- Solve system $(AY)^T AYy = (AY)^T v_0$ (e.g. Gauss) $\quad spec(A) \approx \{0.01, 0.03, 0.56, 0.08, 0.10\}$
- **Solution**: $s(A) = 11232 \cdot A^2 - \frac{38767}{19} \cdot A + \frac{6574}{65} I$. $\quad spec(s(A)A) \approx \{0.64, 0.76, 1.48, 1.25, 1.07\}$



- **Circle Eigenvalue Matrix**
- $b \sim N(0,1)$
- GMRES(100)
- 5000 Iterations
- $\epsilon = 10^{-8}$
  - Gradual improvement with raised degree



- **Circle Eigenvalue Matrix**
- $K(s(A)A)$ (estimated)
- 1 order of magnitude improvement



- **E20R0100**
- **Counter-example for classical preconditioners**
- $b \sim N(0,1)$
- GMRES(100)
- 5000 Iterations
- $\epsilon = 10^{-8}$
  - Achieved by deg 10 polynomial

References

- Iterative Methods for Sparse Linear Systems
  https://www-users.cs.umn.edu/~saad/IterMethBook_2ndEd.pdf
- Templates for the Solutions of Linear Systems: Building Blocks for Iterative Methods
  https://www.netlib.org/templates/templates.pdf
- Polynomial Preconditioned GMRES to Reduce Communication in Parallel Computing
  https://arxiv.org/abs/1907.00072
- Seminar report 😊
  https://github.com/fvanmaele/polynomial-preconditioning

Thank you for your attention! 🤗