70 years of Krylov subspace methods

Erin Carson
Charles University

Collaborators: Jörg Liesen, Zdeněk Strakoš

Trinity College, Dublin
January 25, 2023

Happy Birthday, CG!

Journal of Research of the National Bureau of Standards Vol. 49, No. 6, December 1952 Research Paper 2379

Methods of Conjugate Gradients for Solving Linear Systems[1]

Magnus R. Hestenes [2] and Eduard Stiefel [3]

An iterative algorithm is given for solving a system $Ax=k$ of n linear equations in n unknowns. The solution is given in n steps. It is shown that this method is a special case of a very general method which also includes Gaussian elimination. These general algorithms are essentially algorithms for finding an n dimensional ellipsoid. Connections are made with the theory of orthogonal polynomials and continued fractions.

Krylov subspace methods

- **Linear systems** $Ax = b$, eigenvalue problems, singular value problems, least squares, etc.
- Best for: A large & very sparse, stored implicitly, or only approximation needed

- **Krylov Subspace Method** is a projection process onto the Krylov subspace

$$\mathcal{K}_k(A, r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, \ldots, A^{k-1} r_0\}$$

where A is an $N \times N$ matrix and $r_0 = b - Ax_0$ is a length-N vector

Krylov subspace methods

- **Linear systems** $Ax = b$, eigenvalue problems, singular value problems, least squares, etc.
- Best for: A large & very sparse, stored implicitly, or only approximation needed

- **Krylov Subspace Method** is a projection process onto the Krylov subspace

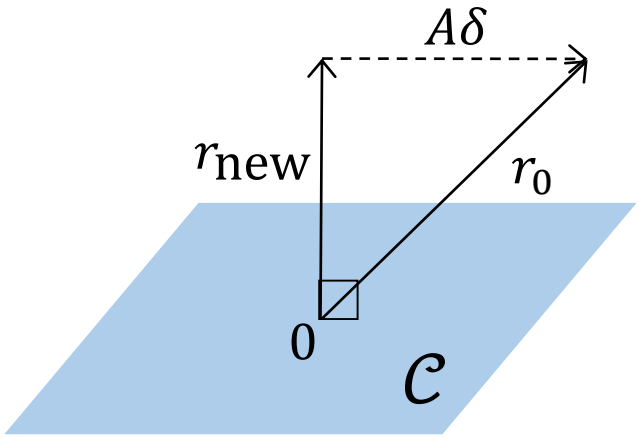
$$\mathcal{K}_k(A, r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, \ldots, A^{k-1} r_0\}$$

where A is an $N \times N$ matrix and $r_0 = b - Ax_0$ is a length-N vector

- In each iteration,
  - Add a dimension to the Krylov subspace
    - Forms nested sequence of Krylov subspaces

$$\mathcal{K}_1(A, r_0) \subset \mathcal{K}_2(A, r_0) \subset \cdots \subset \mathcal{K}_k(A, r_0)$$

  - Orthogonalize (with respect to some $\mathcal{C}_k$)
  - Select approximate solution $x_k \in x_0 + \mathcal{K}_k(A, r_0)$ using $r_k = b - Ax_k \perp \mathcal{C}_k$



Krylov subspace methods

- **Linear systems** $Ax = b$, eigenvalue problems, singular value problems, least squares, etc.
- Best for: A large & very sparse, stored implicitly, or only approximation needed

- **Krylov Subspace Method** is a projection process onto the Krylov subspace

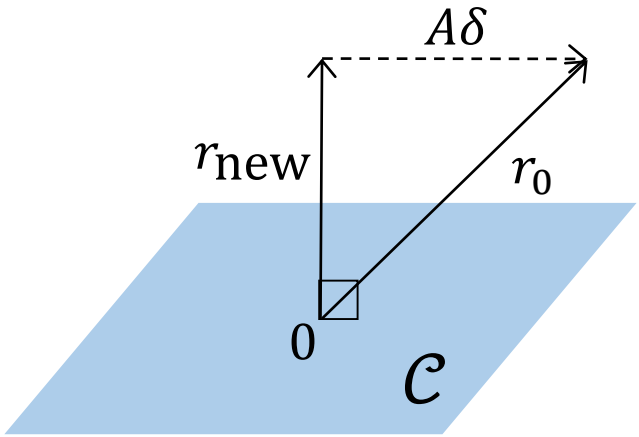
$$\mathcal{K}_k(A, r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, \ldots, A^{k-1} r_0\}$$

where A is an $N \times N$ matrix and $r_0 = b - Ax_0$ is a length-N vector

- In each iteration,
  - Add a dimension to the Krylov subspace
    - Forms nested sequence of Krylov subspaces

$$\mathcal{K}_1(A, r_0) \subset \mathcal{K}_2(A, r_0) \subset \cdots \subset \mathcal{K}_k(A, r_0)$$

  - Orthogonalize (with respect to some $\mathcal{C}_k$)
  - Select approximate solution $x_k \in x_0 + \mathcal{K}_k(A, r_0)$ using $r_k = b - Ax_k \perp \mathcal{C}_k$



- Ex: Lanczos/**Conjugate Gradient (CG)**, Arnoldi/Generalized Minimum Residual (GMRES), Biconjugate Gradient (BICG), BICGSTAB, GKL, LSQR, etc.

The conjugate gradient method

A is symmetric positive definite, $\mathcal{C}_k = \mathcal{K}_k(A, r_0)$

The conjugate gradient method

A is symmetric positive definite, $\mathcal{C}_k = \mathcal{K}_k(A, r_0)$

$$r_k \perp \mathcal{K}_k(A, r_0) \quad \Leftrightarrow \quad x_k = \underset{x_k \in x_0 + \mathcal{K}_k(A, r_0)}{\operatorname{argmin}} \|x - x_k\|_A$$

The conjugate gradient method

A is symmetric positive definite, $\mathcal{C}_k = \mathcal{K}_k(A, r_0)$

$$r_k \perp \mathcal{K}_k(A, r_0) \quad \Leftrightarrow \quad x_k = \underset{x_k \in x_0 + \mathcal{K}_k(A, r_0)}{\operatorname{argmin}} \|x - x_k\|_A$$

$$\Rightarrow \quad r_{N+1} = 0$$

The conjugate gradient method

A is symmetric positive definite, $\mathcal{C}_k = \mathcal{K}_k(A, r_0)$

$$r_k \perp \mathcal{K}_k(A, r_0) \quad \Leftrightarrow \quad x_k = \underset{x_k \in x_0 + \mathcal{K}_k(A, r_0)}{\operatorname{argmin}} \|x - x_k\|_A$$

$$\Rightarrow \quad r_{N+1} = 0$$

Connection with Lanczos

- With $v_1 = r_0/\|r_0\|$, k iterations of Lanczos produces $N \times k$ matrix $V_k = [v_1, \ldots, v_k]$, and $k \times k$ tridiagonal matrix T_k such that
$$AV_k = V_k T_k + \delta_{k+1} v_{k+1} e_k^T, \qquad T_k = V_k^* A V_k$$
- CG approximation x_k is obtained by solving the reduced model
$$T_k y_k = \|r_0\| e_1, \qquad x_k = x_0 + V_k y_k$$

The conjugate gradient method

A is symmetric positive definite, $\mathcal{C}_k = \mathcal{K}_k(A, r_0)$

$$r_k \perp \mathcal{K}_k(A, r_0) \quad \Leftrightarrow \quad x_k = \underset{x_k \in x_0 + \mathcal{K}_k(A, r_0)}{\operatorname{argmin}} \|x - x_k\|_A$$

$$\Longrightarrow \quad r_{N+1} = 0$$

Connection with Lanczos

- With $v_1 = r_0/\|r_0\|$, k iterations of Lanczos produces $N \times k$ matrix $V_k = [v_1, \ldots, v_k]$, and $k \times k$ tridiagonal matrix T_k such that
$$AV_k = V_k T_k + \delta_{k+1} v_{k+1} e_k^T, \qquad T_k = V_k^* A V_k$$
- CG approximation x_k is obtained by solving the reduced model
$$T_k y_k = \|r_0\| e_1, \qquad x_k = x_0 + V_k y_k$$
- Connections with orthogonal polynomials, Stieltjes problem of moments, Gauss-Cristoffel quadrature, others (see 2013 book of Liesen and Strakoš)

The conjugate gradient method

A is symmetric positive definite, $\mathcal{C}_k = \mathcal{K}_k(A, r_0)$

$$r_k \perp \mathcal{K}_k(A, r_0) \quad \Leftrightarrow \quad x_k = \underset{x_k \in x_0 + \mathcal{K}_k(A, r_0)}{\operatorname{argmin}} \|x - x_k\|_A$$

$$\Rightarrow \quad r_{N+1} = 0$$

Connection with Lanczos

- With $v_1 = r_0/\|r_0\|$, k iterations of Lanczos produces $N \times k$ matrix $V_k = [v_1, \ldots, v_k]$, and $k \times k$ tridiagonal matrix T_k such that
$$AV_k = V_k T_k + \delta_{k+1} v_{k+1} e_k^T, \qquad T_k = V_k^* A V_k$$
- CG approximation x_k is obtained by solving the reduced model
$$T_k y_k = \|r_0\| e_1, \qquad x_k = x_0 + V_k y_k$$
- Connections with orthogonal polynomials, Stieltjes problem of moments, Gauss-Cristoffel quadrature, others (see 2013 book of Liesen and Strakoš)

⇒ CG (and other Krylov subspace methods) are highly nonlinear

- Good for convergence, bad for ease of finite precision analysis

Hestenes and Stiefel CG Algorithm

- Uses three 2-term recurrences for updating $x_{k+1}, r_{k+1}, p_{k+1}$

$r_0 = b - Ax_0, \;\; p_0 = r_0$

for $k = 0$:nmax

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k A p_k$$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$p_{k+1} = r_{k+1} + \beta_{k+1} p_k$$

end

Hestenes and Stiefel CG Algorithm

- Uses three 2-term recurrences for updating $x_{k+1}, r_{k+1}, p_{k+1}$

$r_0 = b - Ax_0, \quad p_0 = r_0$

for $k = 0$:nmax

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$$

minimizes $\|x - x_k\|_A$ along line $z(\alpha) = x_k + \alpha p_k$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k A p_k$$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$p_{k+1} = r_{k+1} + \beta_{k+1} p_k$$

end

Hestenes and Stiefel CG Algorithm

- Uses three 2-term recurrences for updating $x_{k+1}, r_{k+1}, p_{k+1}$

$r_0 = b - Ax_0, \quad p_0 = r_0$
for $k = 0$:nmax

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k A p_k$$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$p_{k+1} = r_{k+1} + \beta_{k+1} p_k$$

end

minimizes $\|x - x_k\|_A$ along line $z(\alpha) = x_k + \alpha p_k$

If

$$p_k \perp_A p_j \text{ for } k \neq j,$$

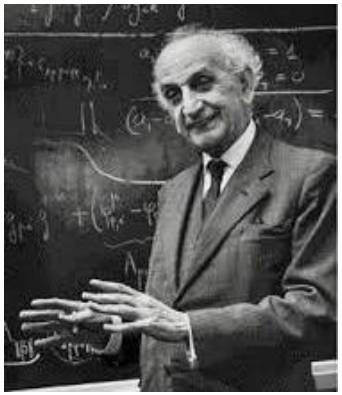
1-dimensional minimizations in each iteration give k-dimensional minimization over the whole subspace

$$x_0 + \mathcal{K}_k(A, r_0) = x_0 + \text{span}\{p_0, \ldots p_{k-1}\}$$

CG History

- Institute of Numerical Analysis (INA), founded by the US National Bureau of Standards on the campus of UCLA

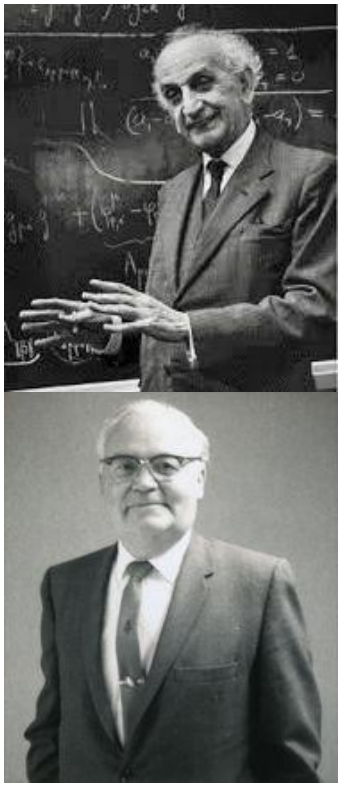
- Lanczos employed as a member of the research staff



CG History

- Institute of Numerical Analysis (INA), founded by the US National Bureau of Standards on the campus of UCLA

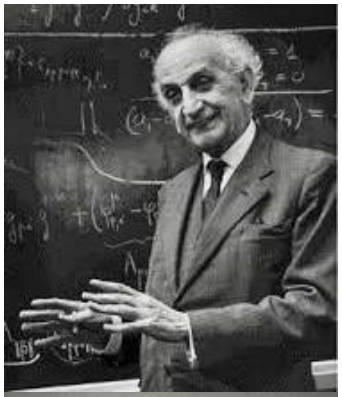
- Lanczos employed as a member of the research staff
- Hestenes was a professor at UCLA
  - Developed an iterative method for solving SPD linear systems in June-July 1951



CG History

- Institute of Numerical Analysis (INA), founded by the US National Bureau of Standards on the campus of UCLA

- Lanczos employed as a member of the research staff
- Hestenes was a professor at UCLA
  - Developed an iterative method for solving SPD linear systems in June-July 1951
- Stiefel was a professor at ETH Zurich
  - Came to INA for a symposium in August 1951, planning to give a talk about an iterative method for solving SPD linear systems



Mathematical Properties of CG

Let $A = Q\Lambda Q^T$, and let $r_0 = Q[\eta_1, \ldots, \eta_N]^T$

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \|p(A)(x - x_0)\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2}$$

Mathematical Properties of CG

Let $A = Q\Lambda Q^T$, and let $r_0 = Q[\eta_1, \ldots, \eta_N]^T$

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \|p(A)(x - x_0)\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2}$$

Let $\theta_1^{(k)}, \ldots, \theta_k^{(k)}$ be the k roots of the polynomial providing the minimum above. Then

$$\|x - x_k\|_A^2 = \sum_{i=1}^{N} \prod_{\ell=1}^{k} \left(1 - \frac{\lambda_i}{\theta_\ell^{(k)}} \right)^2 \frac{\eta_i^2}{\lambda_i}$$

CG Convergence Bounds

$$\sum_{i=1}^{N} \frac{\eta_i^2}{\lambda_i} = r_0^T A^{-1} r_0 = \|x - x_0\|_A^2$$

CG Convergence Bounds

$$\sum_{i=1}^{N} \frac{\eta_i^2}{\lambda_i} = r_0^T A^{-1} r_0 = \|x - x_0\|_A^2$$

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2} \Rightarrow \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in P_k(0)} \max_{1 \leq i \leq N} |p(\lambda_i)|$$

CG Convergence Bounds

$$\sum_{i=1}^{N} \frac{\eta_i^2}{\lambda_i} = r_0^T A^{-1} r_0 = \|x - x_0\|_A^2$$

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2} \Rightarrow \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in P_k(0)} \max_{1 \leq i \leq N} |p(\lambda_i)|$$

"polynomial min-max approximation bound"

Note: does not depend on r_0!

CG Convergence Bounds

$$\sum_{i=1}^{N} \frac{\eta_i^2}{\lambda_i} = r_0^T A^{-1} r_0 = \|x - x_0\|_A^2$$

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2} \Rightarrow \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in P_k(0)} \max_{1 \leq i \leq N} |p(\lambda_i)|$$

"polynomial min-max approximation bound"
Note: does not depend on r_0!

Further simplification: Replace $\{\lambda_1, \ldots, \lambda_N\}$ by continuous interval $[\lambda_1, \lambda_N]$ and use Chebyshev polynomials on this interval:

$$\min_{p \in P_k(0)} \max_{1 \leq i \leq N} |p(\lambda_i)| \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \Rightarrow \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k$$

CG Convergence Bounds

$$\sum_{i=1}^{N} \frac{\eta_i^2}{\lambda_i} = r_0^T A^{-1} r_0 = \|x - x_0\|_A^2$$

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2} \Rightarrow \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \min_{p \in P_k(0)} \max_{1 \leq i \leq N} |p(\lambda_i)|$$

"polynomial min-max approximation bound"
Note: does not depend on r_0!

Further simplification: Replace $\{\lambda_1, \ldots, \lambda_N\}$ by continuous interval $[\lambda_1, \lambda_N]$ and use Chebyshev polynomials on this interval:

$$\min_{p \in P_k(0)} \max_{1 \leq i \leq N} |p(\lambda_i)| \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \Rightarrow \frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k$$

"$\kappa(A)$-bound"
Note: does not depend on r_0 OR eigenvalue distribution!

Diagonal test problems

For given $N \geq 3$, $0 < \lambda_1 < \lambda_N$, and $\rho > 0$, define

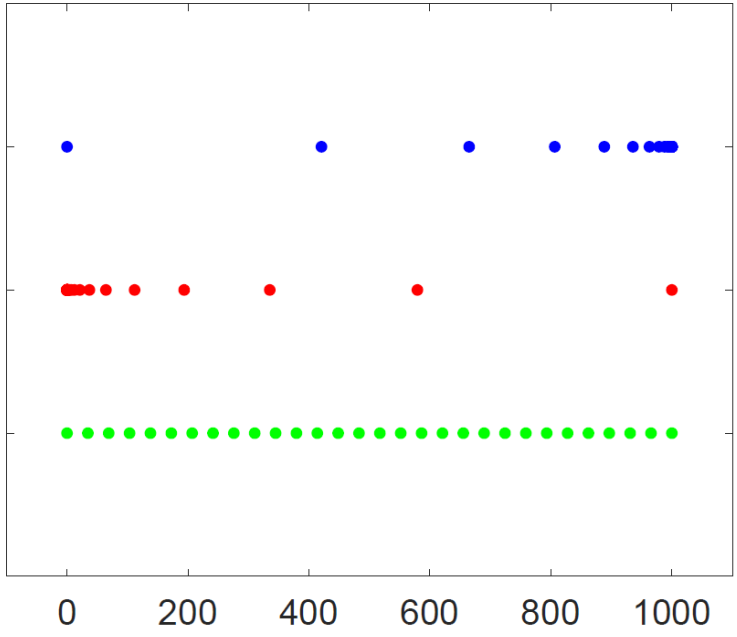
$$A = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{N-1}, \lambda_N) \quad \text{where} \quad \lambda_i = \lambda_1 + \left(\frac{i-1}{N-1}\right)(\lambda_N - \lambda_1)\rho^{N-i}$$

for $i = 2, \ldots, N-1$.

Example 1: The mathematical behavior of CG for different eigenvalue distributions

Setup:

1. Modified diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - reversed so eigenvalues accumulate on the right side of the spectrum
2. Diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - eigenvalues accumulated to the left side of the spectrum
3. Diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 1.0$
   - Equally-spaced eigenvalues



Example 1: The mathematical behavior of CG for different eigenvalue distributions

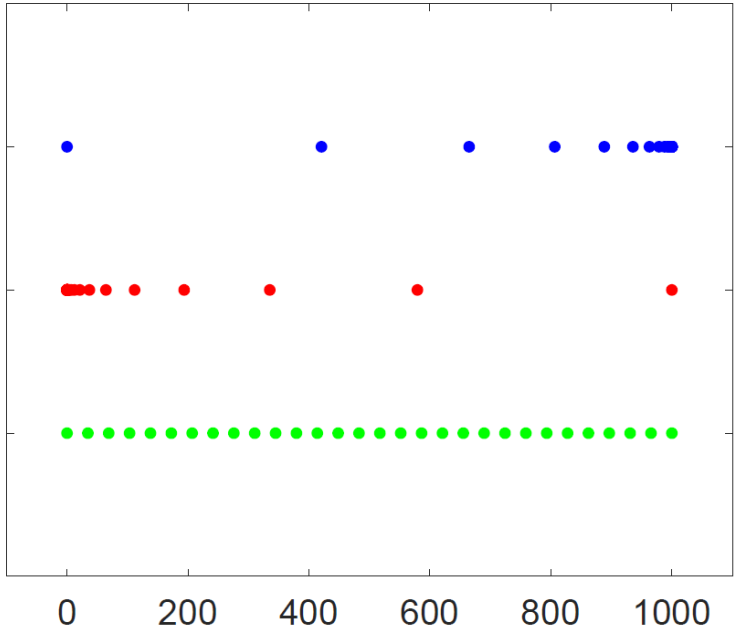
Setup:

1. Modified diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - reversed so eigenvalues accumulate on the right side of the spectrum
2. Diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - eigenvalues accumulated to the left side of the spectrum
3. Diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 1.0$
   - Equally-spaced eigenvalues

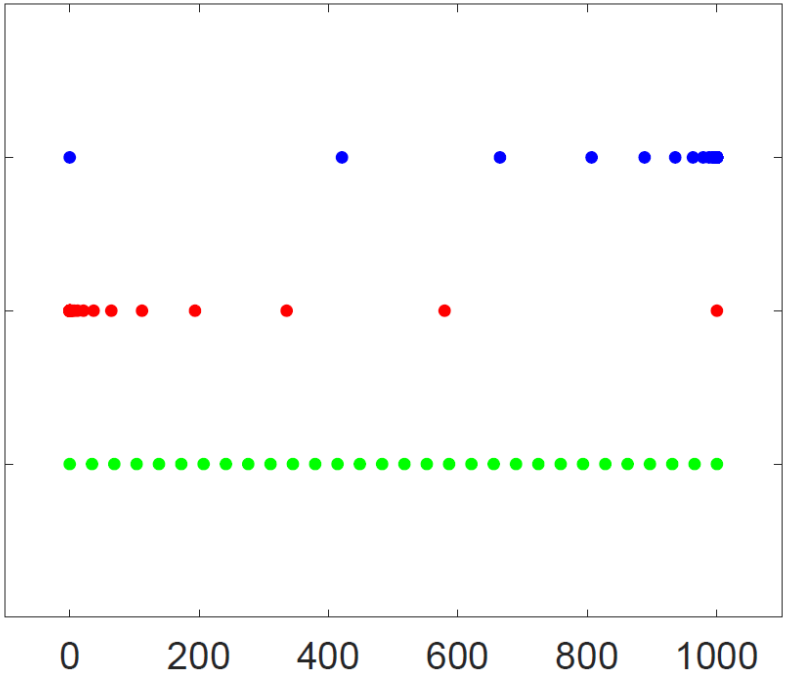
Run exact CG on these three matrices with:

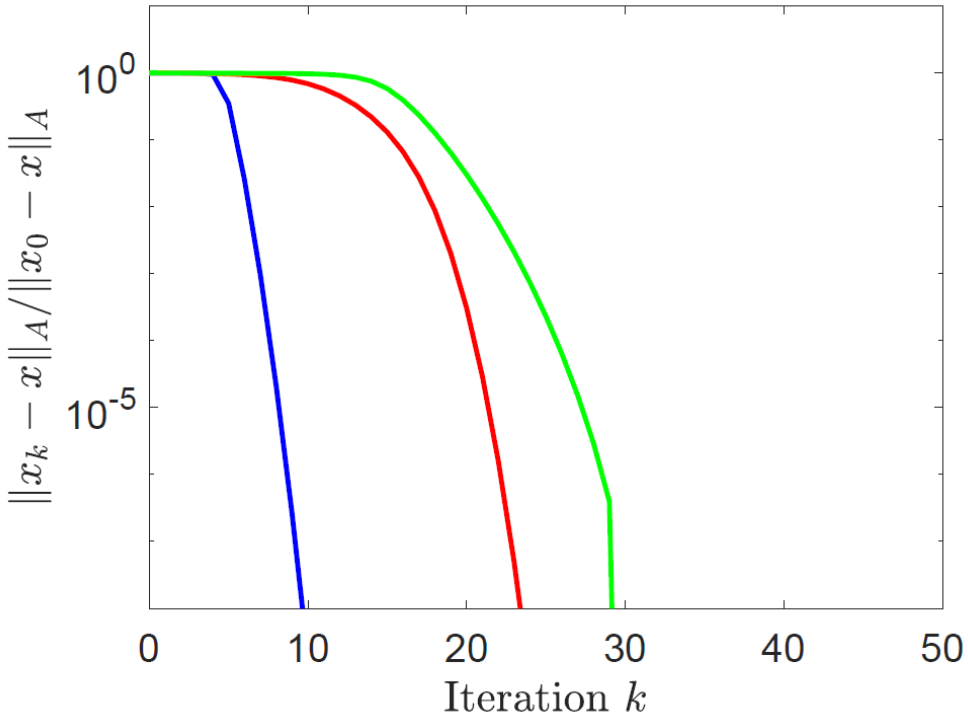
$b = [1, \ldots, 1]^T/\sqrt{N}$

$x_0 = 0$



Example 1: The mathematical behavior of CG for different eigenvalue distributions





Example 1: The mathematical behavior of CG for different eigenvalue distributions

Explanation:

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \|p(A)(x - x_0)\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2}$$

For eigenvalues accumulated to the right:

- Roots of CG polynomial approximate small outlying eigenvalues within a few iterations
- The rest of eigenvalues are large and close, so values of the polynomial do not need to be small, since their squares are divided by large eigenvalues
- This means fast convergence will occur

Example 1: The mathematical behavior of CG for different eigenvalue distributions

Explanation:

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \|p(A)(x - x_0)\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2}$$

For eigenvalues accumulated to the left:

- Roots of CG polynomial approximate large outlying eigenvalues within a few iterations
- BUT for the bulk of the small eigenvalues, the CG polynomial must place many roots close to the left end of the spectrum to make up for the division of its squares values by the small eigenvalues
- This means acceleration occurs much later

Example 1: The mathematical behavior of CG for different eigenvalue distributions

Explanation:

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \|p(A)(x - x_0)\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2}$$

For equally-spaced eigenvalues:

- Roots of CG polynomial slowly approximate individual eigenvalues, which proceeds from both edges of the spectrum

Example 1: The mathematical behavior of CG for different eigenvalue distributions

Main point: The convergence of CG depends on the eigenvalue distribution.

Example 2: Worst case CG and the quality of convergence bounds

Setup:

4 linear systems:

1. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 5,\ \rho = 1.0$; RHS $b = [1, \ldots, 1]^T/\sqrt{N}$

   equally spaced eigenvalues, small condition number, RHS w/equal comps in eigenbasis

Example 2: Worst case CG and the quality of convergence bounds

Setup:

4 linear systems:

1. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 5,\ \rho = 1.0$; RHS $b = [1, \ldots, 1]^T/\sqrt{N}$

   equally spaced eigenvalues, small condition number, RHS w/equal comps in eigenbasis

2. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 100,\ \rho = 1.0$; RHS $b = [1, \ldots, 1]^T/\sqrt{N}$

   equally spaced eigenvalues, **larger condition number**, RHS w/equal comps in eigenbasis

Example 2: Worst case CG and the quality of convergence bounds

Setup:

4 linear systems:

1. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 5,\ \rho = 1.0$; RHS $b = [1, \ldots, 1]^T/\sqrt{N}$

   equally spaced eigenvalues, small condition number, RHS w/equal comps in eigenbasis

2. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 100,\ \rho = 1.0$; RHS $b = [1, \ldots, 1]^T/\sqrt{N}$

   equally spaced eigenvalues, **larger condition number**, RHS w/equal comps in eigenbasis

3. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 5,\ \rho = 0.1$; RHS $b = [1, \ldots, 1]^T/\sqrt{N}$

   eigenvalues accumulated to the left, small condition number, RHS w/equal comps in eigenbasis

Example 2: Worst case CG and the quality of convergence bounds

Setup:

4 linear systems:

1. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 5,\ \rho = 1.0$; RHS $b = [1, \dots, 1]^T/\sqrt{N}$

   equally spaced eigenvalues, small condition number, RHS w/equal comps in eigenbasis

2. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 100,\ \rho = 1.0$; RHS $b = [1, \dots, 1]^T/\sqrt{N}$

   equally spaced eigenvalues, **larger condition number**, RHS w/equal comps in eigenbasis

3. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 5,\ \rho = 0.1$; RHS $b = [1, \dots, 1]^T/\sqrt{N}$

   eigenvalues accumulated to the left, small condition number, RHS w/equal comps in eigenbasis

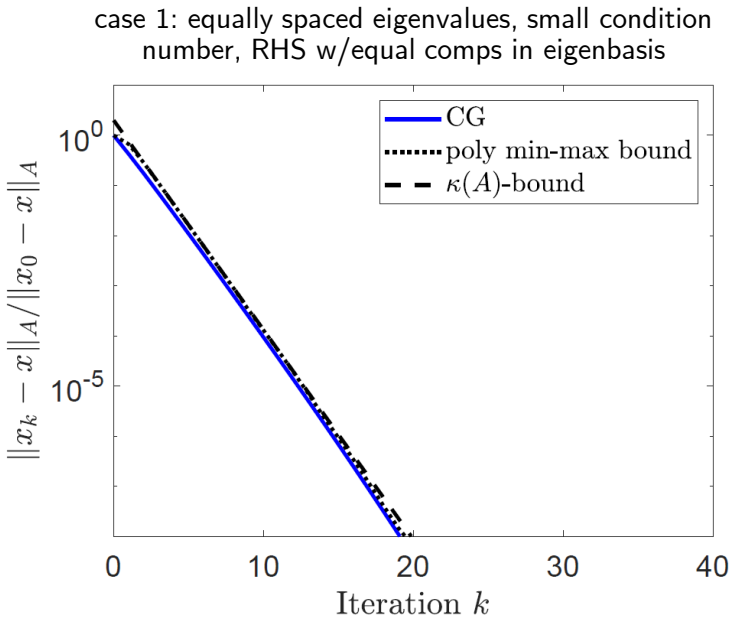
4. Diagonal matrix with $N = 48,\ \lambda_1 = 1,\ \lambda_N = 5,\ \rho = 1.0$; RHS $b = [\eta_1, \dots, \eta_N]^T$ is a unit-norm vector with $|\eta_1|, |\eta_N| \approx 1, |\eta_i| \approx 10^{-13}, i = 2, \dots, N-1$

   equally spaced eigenvalues, small condition number, RHS w/**unequal** comps in eigenbasis

Run exact CG with $x_0 = 0$

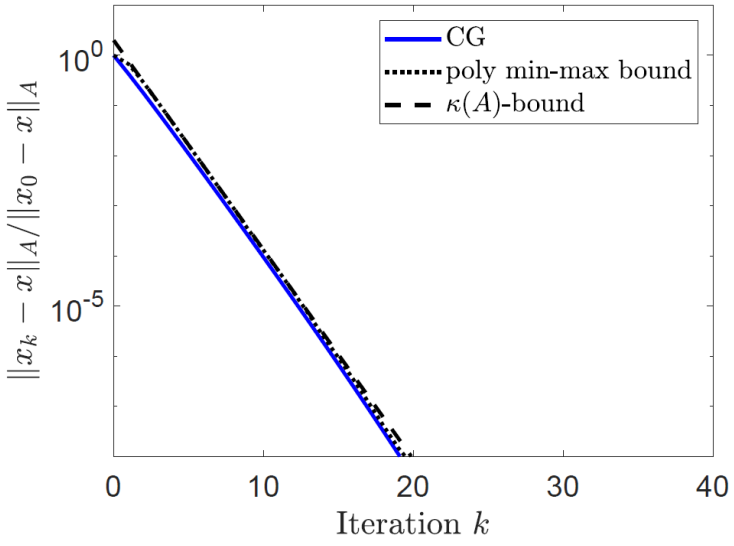
Plot $\kappa(A)$-bound and polynomial min-max approximation bound

Example 2: Worst case CG and the quality of convergence bounds

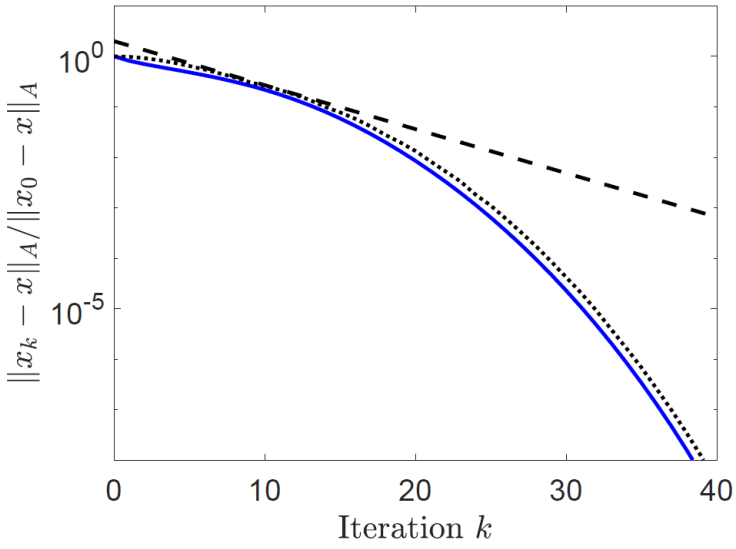


Example 2: Worst case CG and the quality of convergence bounds

case 1: equally spaced eigenvalues, small condition number, RHS w/equal comps in eigenbasis

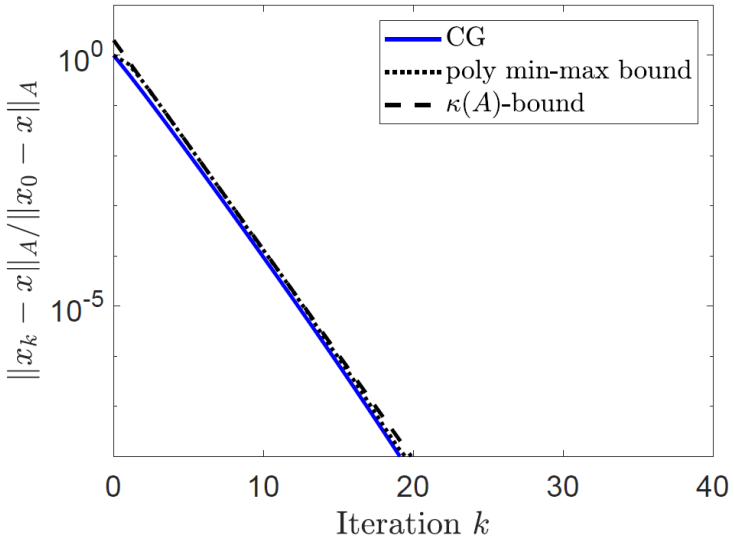


case 2: equally spaced eigenvalues, larger condition number, RHS w/equal comps in eigenbasis

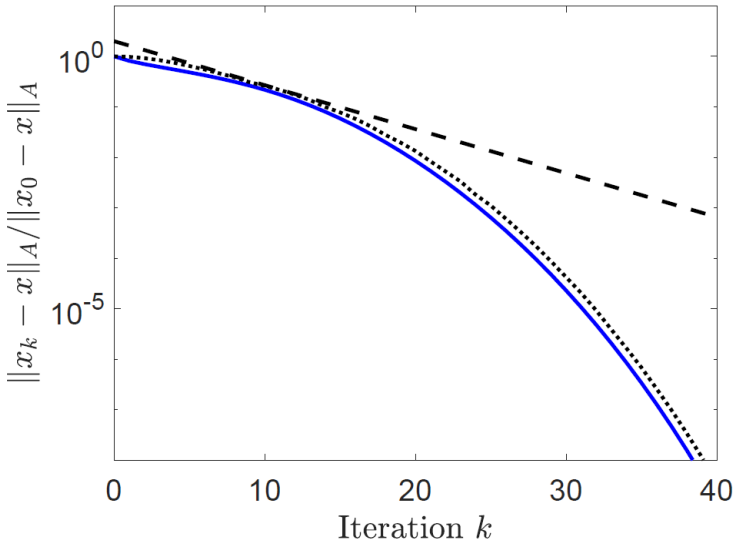


Example 2: Worst case CG and the quality of convergence bounds

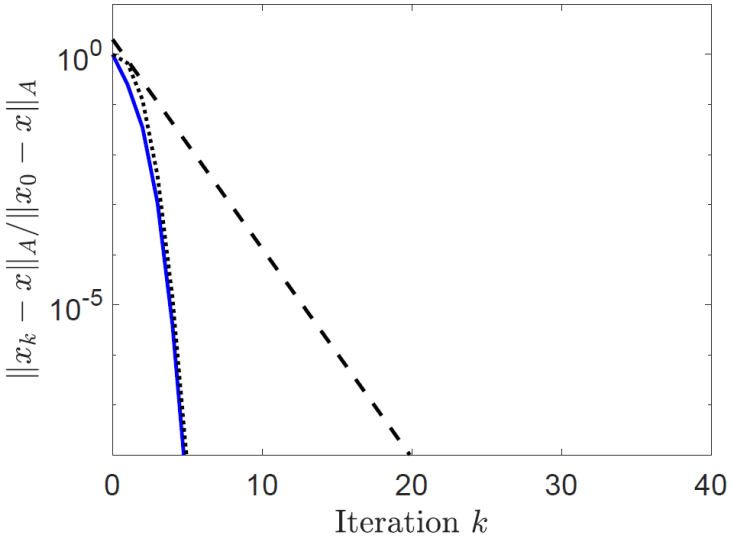
case 1: equally spaced eigenvalues, small condition number, RHS w/equal comps in eigenbasis



case 2: equally spaced eigenvalues, larger condition number, RHS w/equal comps in eigenbasis

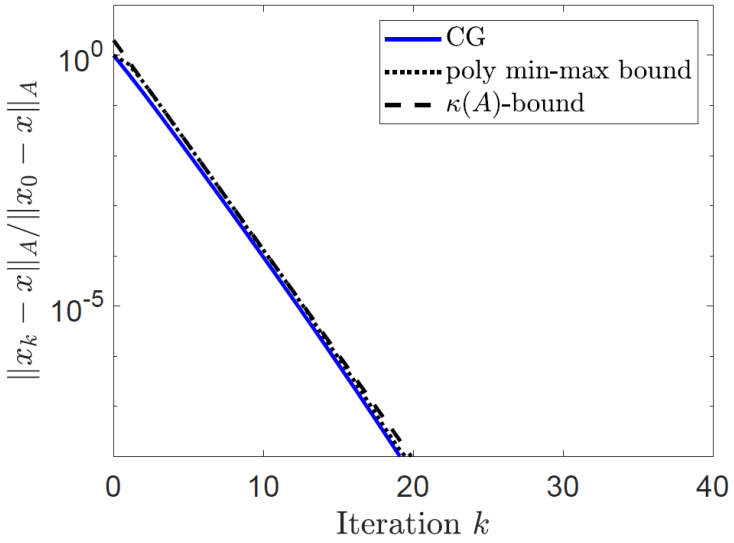


case 3: evals accum. to left, small condition number, RHS w/equal comps in eigenbasis

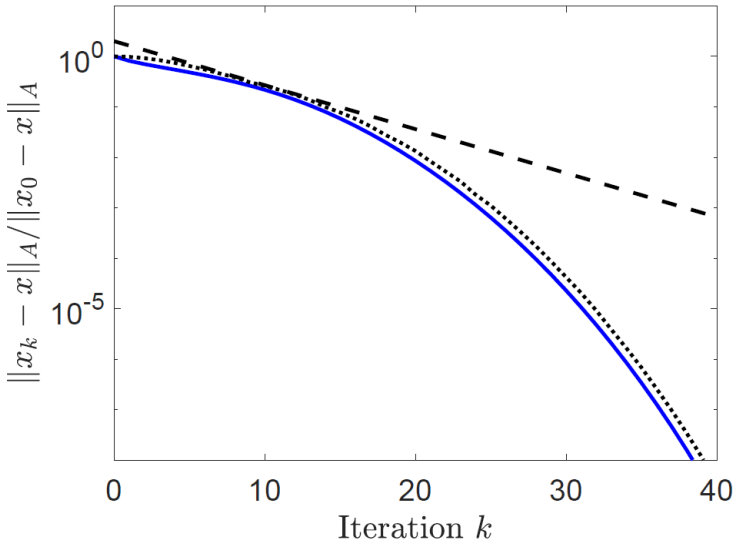


Example 2: Worst case CG and the quality of convergence bounds

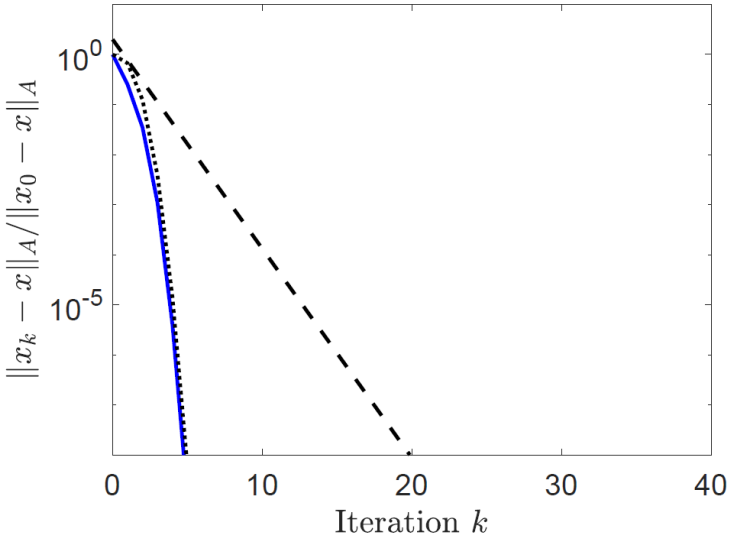
case 1: equally spaced eigenvalues, small condition number, RHS w/equal comps in eigenbasis



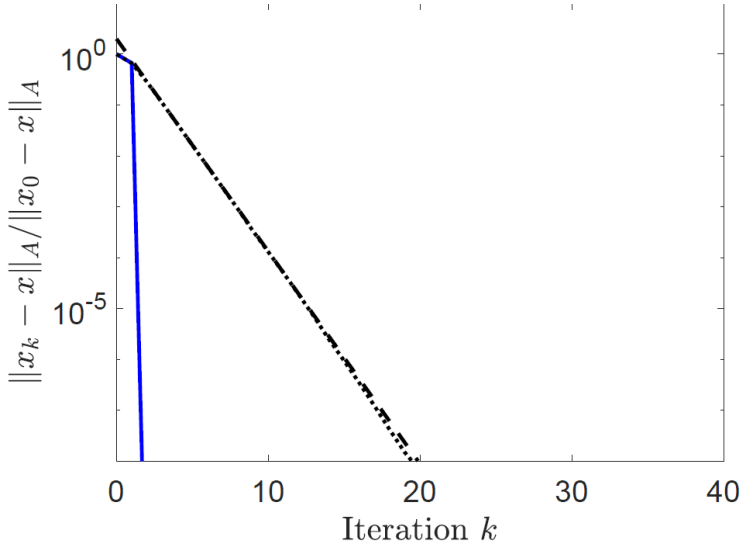
case 2: equally spaced eigenvalues, larger condition number, RHS w/equal comps in eigenbasis



case 3: evals accum. to left, small condition number, RHS w/equal comps in eigenbasis



case 4: equally spaced eigenvalues, small condition number, RHS w/unequal comps in eigenbasis



Example 2: Worst case CG and the quality of convergence bounds

The $\kappa(A)$-bound does not take into account the eigenvalue distribution or the right-hand side, so it is only qualitatively/qualitatively descriptive when:

1. A is well-conditioned
2. The eigenvalues of A are uniformly distributed
3. The right-hand side contains sizeable components in all eigenvectors of A

The polynomial min-max approximation bound takes into account the eigenvalue distribution, but not the right-hand side, so it is only descriptive when:

1. The right-hand side contains sizeable components in all eigenvectors of A

Example 2: Worst case CG and the quality of convergence bounds

Main point: The $\kappa(A)$-bound does not account for the eigenvalue distribution or the initial residual (right-hand side) and thus can be tight only in a very particular case.

Example 3: Numerical behavior of CG on standard model problems

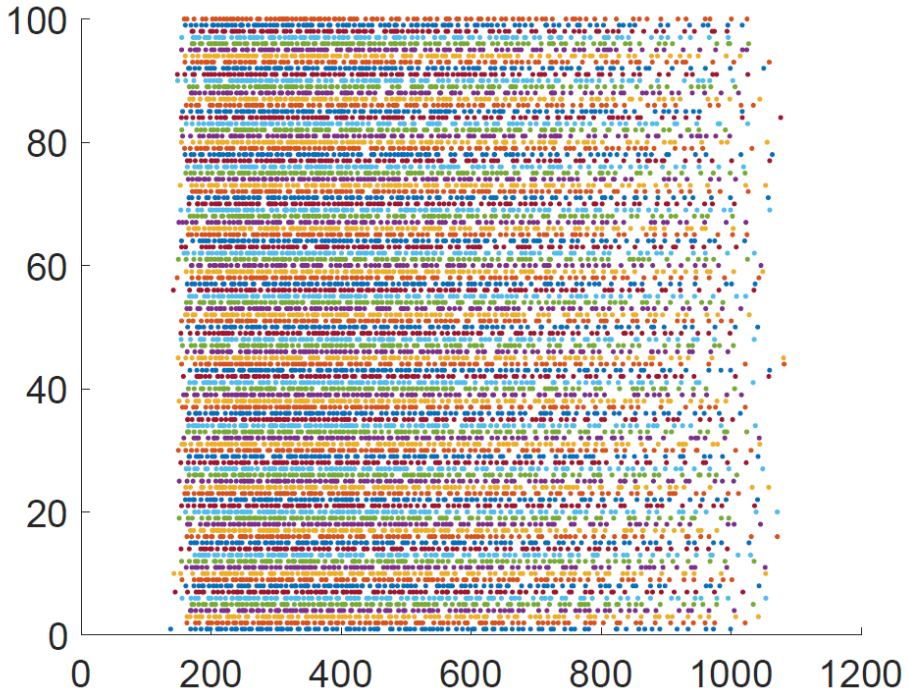
Setup:

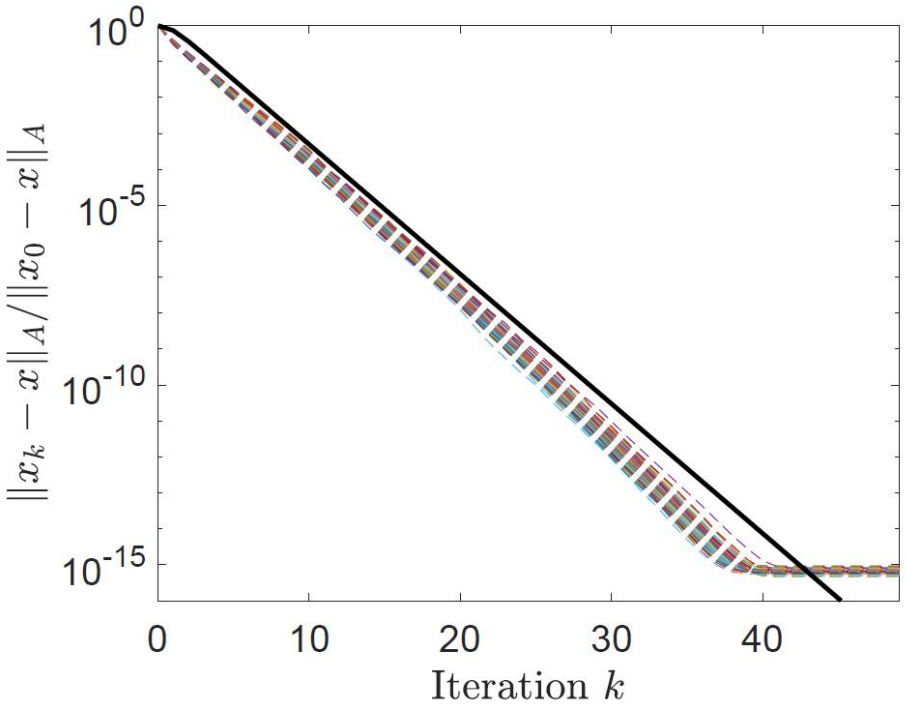
2 commonly-used model problems:

1. Wishart matrices: $A = R^T R$ where $R \in \mathbb{R}^{M \times N}, M \geq N$ is a full-rank random matrix drawn from standard normal distribution
   - $M = 500, N = 100$
2. 2D Poisson problem discretized on 50×50 grid ($N = 2500$)

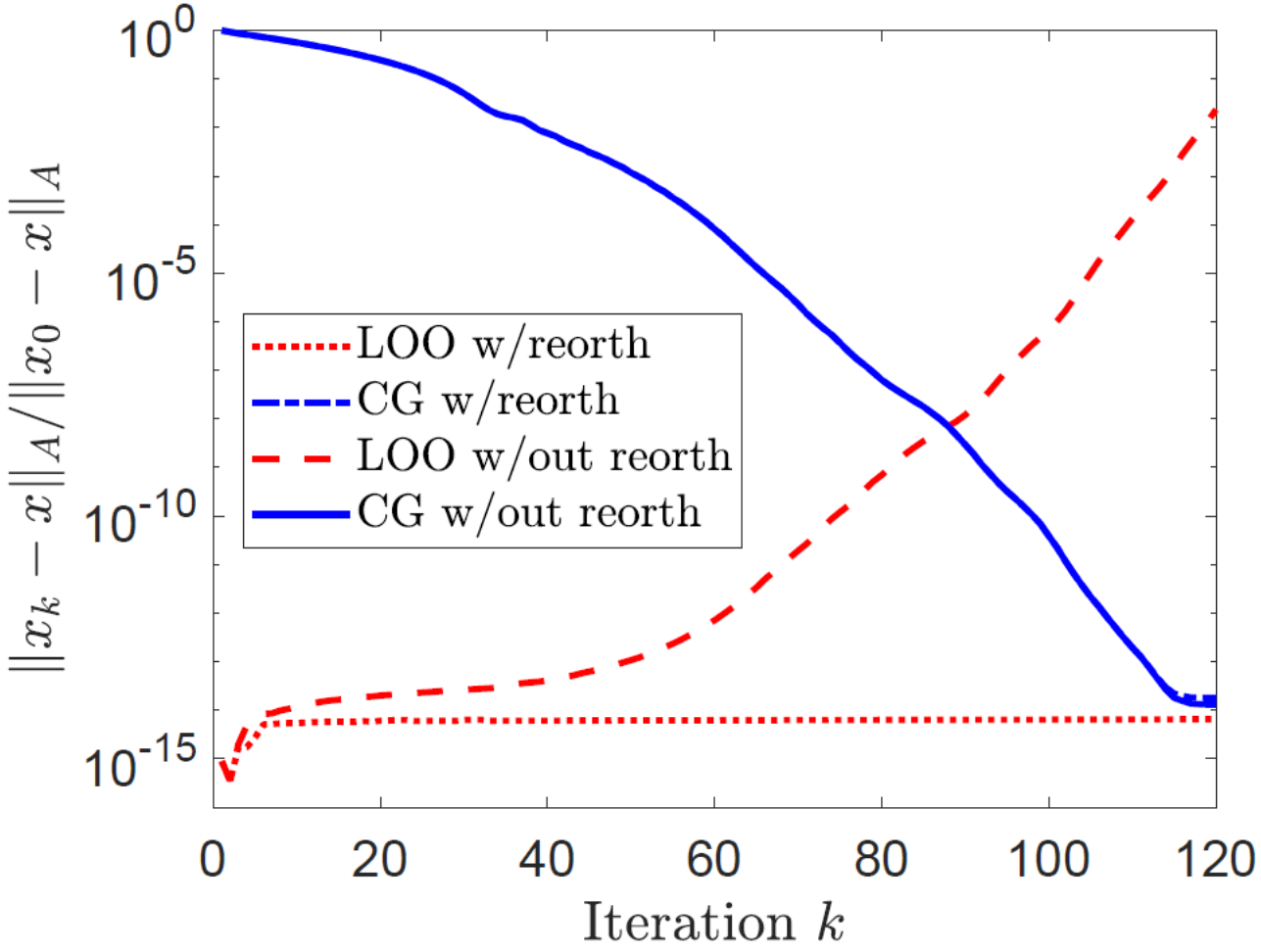
Run CG in double precision, with $b = [1, \ldots, 1]^T / \sqrt{N}$, $x_0 = 0$

Example 3: Numerical behavior of CG on standard model problems





Example 3: Numerical behavior of CG on standard model problems



Example 3: Numerical behavior of CG on standard model problems

- Wishart matrices have a provably low condition number and rather evenly-spaced eigenvalues
- Thus we expect CG will converge quickly

- For Poisson problem, eigenvalues are also rather evenly-spaced
- Loss of orthogonality is very gradual
- Finite precision error does not cause delay of convergence

Example 3: Numerical behavior of CG on standard model problems

Main point: Some model problems typically used for studying the behavior of CG are canonically easy cases for CG and are not indicative of the behavior of CG in general.

Example 4: Preconditioned CG and the condition number

Setup:

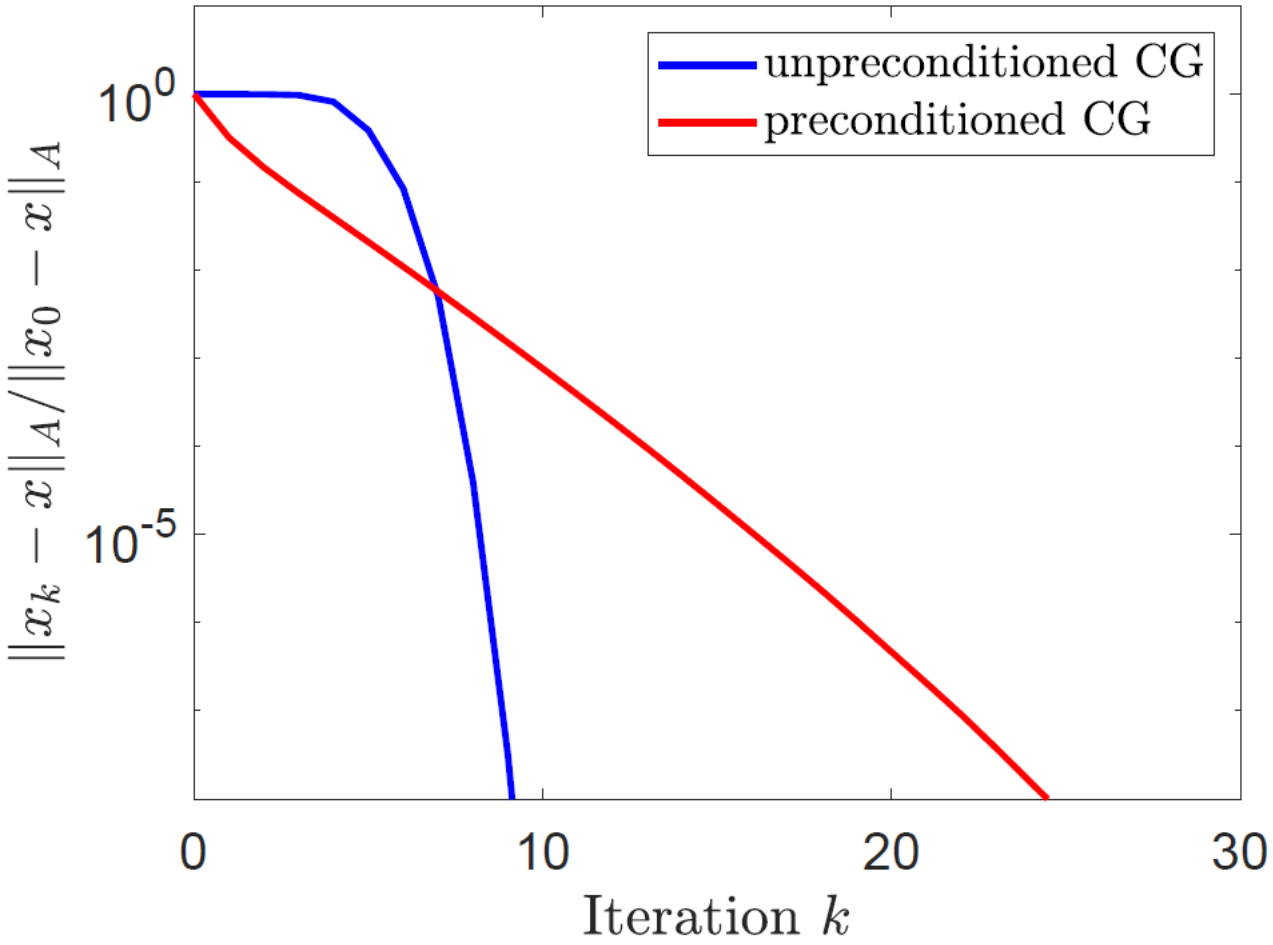
A: Diagonal matrix with $N = 40,\ \lambda_1 = 10^{-3},\ \lambda_N = 100,\ \rho = 0.1;$

$b = [1, \ldots, 1]^T/\sqrt{N}$

Let preconditioner P be a diagonal matrix such that $P^{-1}A$ is diagonal with eigenvalues equally spaced between $\lambda_1 = 10$ and $\lambda_N = 100$.

Run exact CG with $x_0 = 0$.

Example 4: Preconditioned CG and the condition number



Example 4: Preconditioned CG and the condition number

- As we have seen, the distribution of eigenvalues is what matters for CG!
- The goal of preconditioning is NOT to lower the condition number!

[Gergelits, Mardal, Nielsen, Strakoš, SINUM 57, 2019]

Example 4: Preconditioned CG and the condition number

Main point: Smaller condition number $\neq$ faster convergence.

Example 5: Mathematical behavior of CG for problems with clustered eigenvalues

Setup:

1. Modified diagonal matrix with $N = 10,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - reversed so eigenvalues accumulate on the right side of the spectrum
2. Diagonal matrix with $N = 10,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - eigenvalues accumulated to the left side of the spectrum
3. Diagonal matrix with $N = 10,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 1.0$
   - equally-spaced eigenvalues

Example 5: Mathematical behavior of CG for problems with clustered eigenvalues

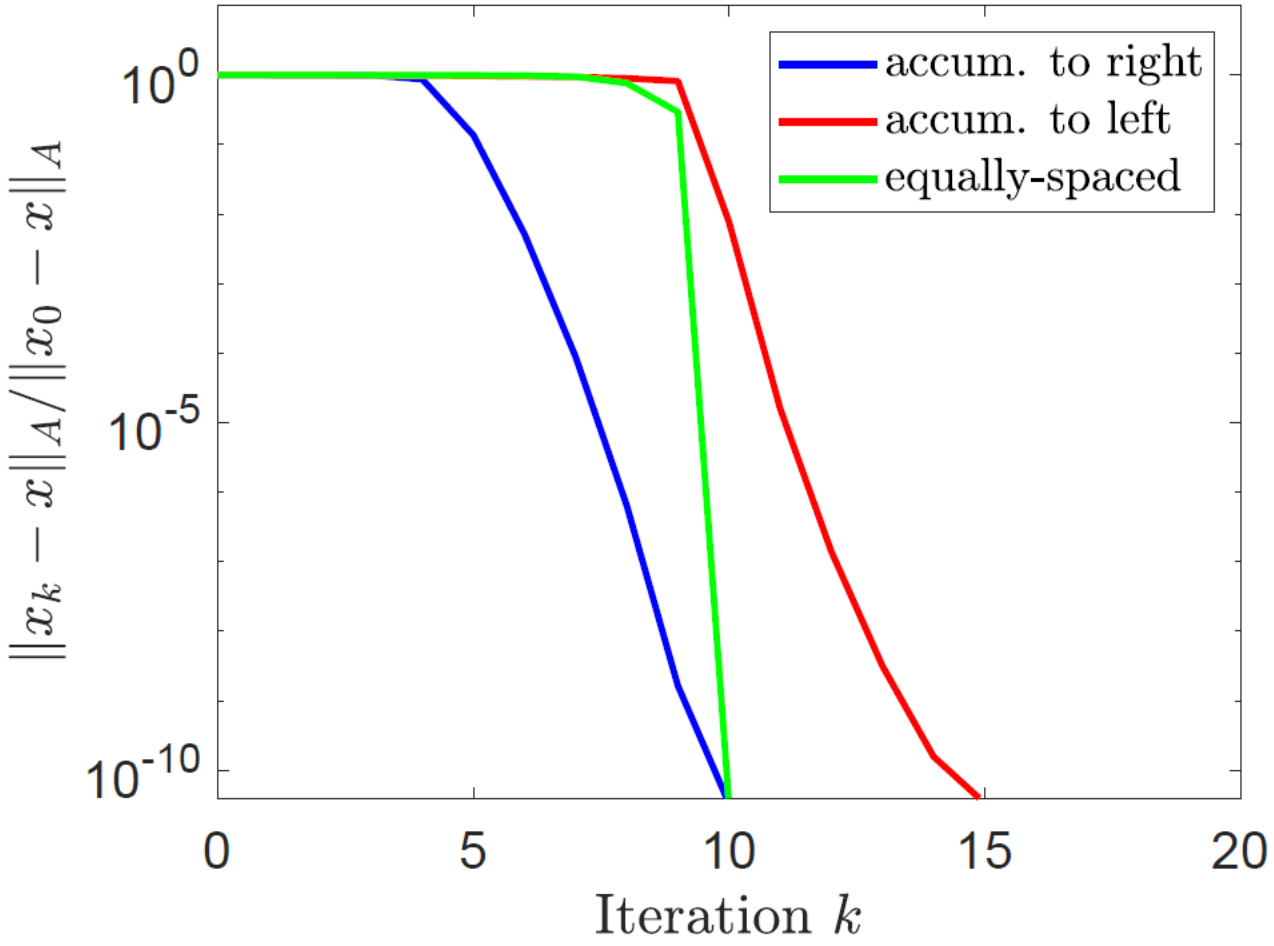
Setup:

1. Modified diagonal matrix with $N = 10,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - reversed so eigenvalues accumulate on the right side of the spectrum
2. Diagonal matrix with $N = 10,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - eigenvalues accumulated to the left side of the spectrum
3. Diagonal matrix with $N = 10,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 1.0$
   - equally-spaced eigenvalues

For each case, construct a matrix of size $N = 100$ by replacing each eigenvalue by a tight cluster of 10 eigenvalues, where clusters have diameter $O(10^{-12})$.

Run exact CG with $b = [1, \ldots, 1]^T/\sqrt{N},\ x_0 = 0.$

Example 5: Mathematical behavior of CG for problems with clustered eigenvalues



Example 5: Mathematical behavior of CG for problems with clustered eigenvalues

- In 10 iterations the CG polynomials (in all cases) place a single Ritz value in each cluster
  - For the clusters accumulated to the right and equally spaced, sufficient for approximating the minimal polynomial

- For the clusters accumulated to the left, one Ritz value per cluster is not enough to decrease CG error
  - To achieve the desired decrease of the error, CG must place additional Ritz values in the rightmost clusters, which delays convergence

[Greenbaum and Strakoš, SIMAX 13, 1992]

Section 5.6 of [Liesen and Strakoš, 2013]

Example 5: Mathematical behavior of CG for problems with clustered eigenvalues

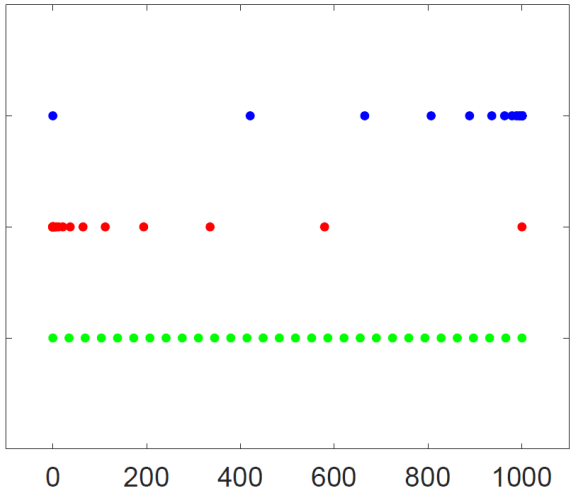
Main point: A spectrum localized in ℓ tight clusters does not mean reaching a good CG approximation to the solution in ℓ steps.

Example 6: Sensitivity of CG to rounding errors

Setup:

1. Modified diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - reversed so eigenvalues accumulate on the right side of the spectrum
2. Diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - eigenvalues accumulated to the left side of the spectrum
3. Diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 1.0$
   - Equally-spaced eigenvalues

Run **double precision** CG on these three matrices with $b = [1, \ldots, 1]^T/\sqrt{N}, x_0 = 0$

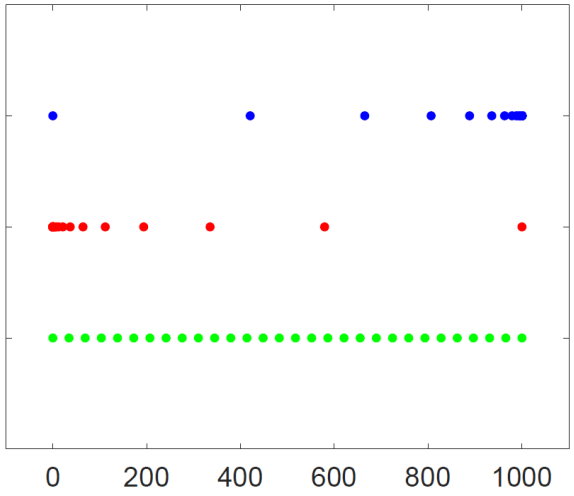


Example 6: Sensitivity of CG to rounding errors

Setup:

1. Modified diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - reversed so eigenvalues accumulate on the right side of the spectrum
2. Diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 0.6$
   - eigenvalues accumulated to the left side of the spectrum
3. Diagonal matrix with $N = 30,\ \lambda_1 = 0.1,\ \lambda_N = 10^3,\ \rho = 1.0$
   - Equally-spaced eigenvalues

Run **double precision** CG on these three matrices with $b = [1, \ldots, 1]^T/\sqrt{N}, x_0 = 0$

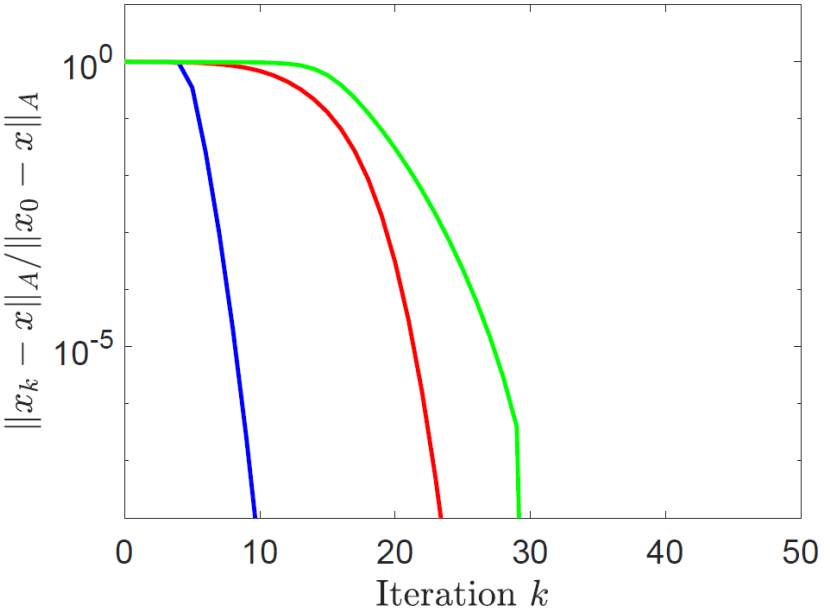


For each matrix, create a larger matrix by replacing each eigenvalue with a cluster of 4 eigenvalues, with cluster diameter $O(10^{-13})$.

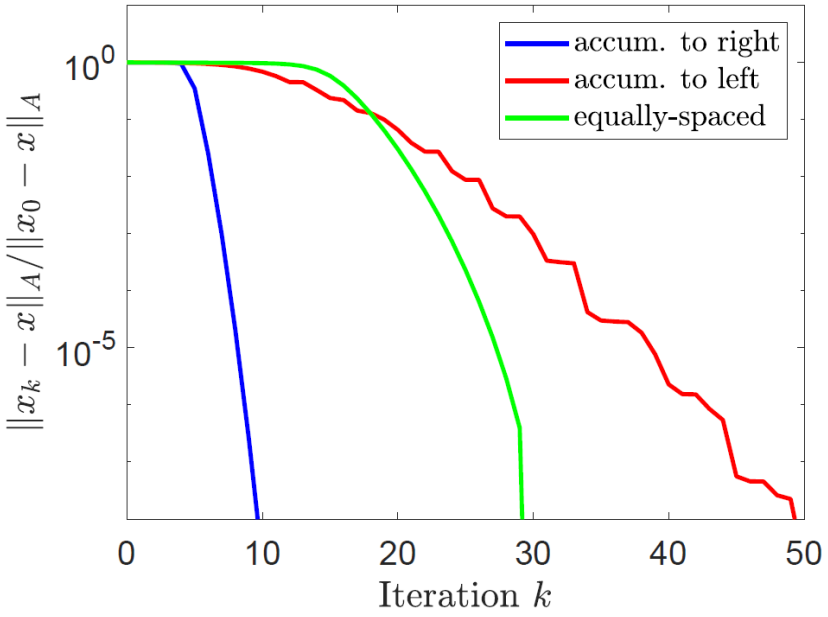
Run **exact** CG on these three matrices with $b = [1, \ldots, 1]^T/\sqrt{N}, x_0 = 0$

Example 6: Sensitivity of CG to rounding errors

exact CG on matrices without clusters

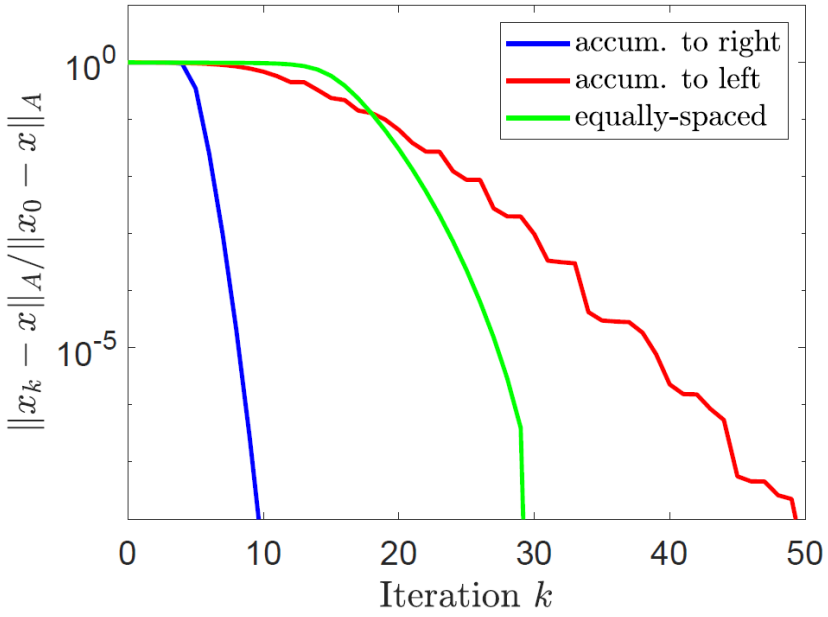


double precision CG on matrices without clusters

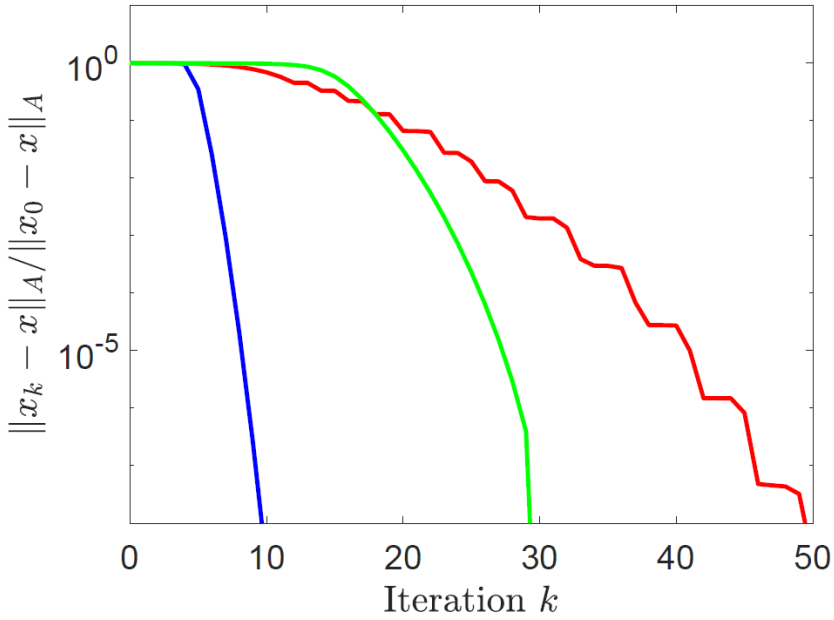


Example 6: Sensitivity of CG to rounding errors

double precision CG on matrices without clusters



exact CG on matrices with clusters



Example 6: Sensitivity of CG to rounding errors

2 phenomena working against each other!

- Large outlying eigenvalues desirable in exact arithmetic
- But cause problem to be more sensitive to rounding errors

[Strakoš, LAA 154, 1991], [Jennings, IMA JAM 20, 1977]

For connection of double precision CG to exact CG on larger matrix with tight clusters, see [Greenbaum, LAA 113, 1989]

Example 6: Sensitivity of CG to rounding errors

Main point: Large outlying eigenvalues cause CG convergence to be more susceptible to delay caused by finite precision errors.

Convergence behavior of finite precision CG can be equated with exact CG on a larger problem whose eigenvalues are replaced by tight clusters.

Example 7: Computational behavior of different CG algorithms

Setup:

A: Diagonal matrix with $N = 48$, $\lambda_1 = 0.1, \lambda_N = 10^3, \rho = 0.25$

$b = [1, \dots, 1]^T / \sqrt{N}\,, x_0 = 0$

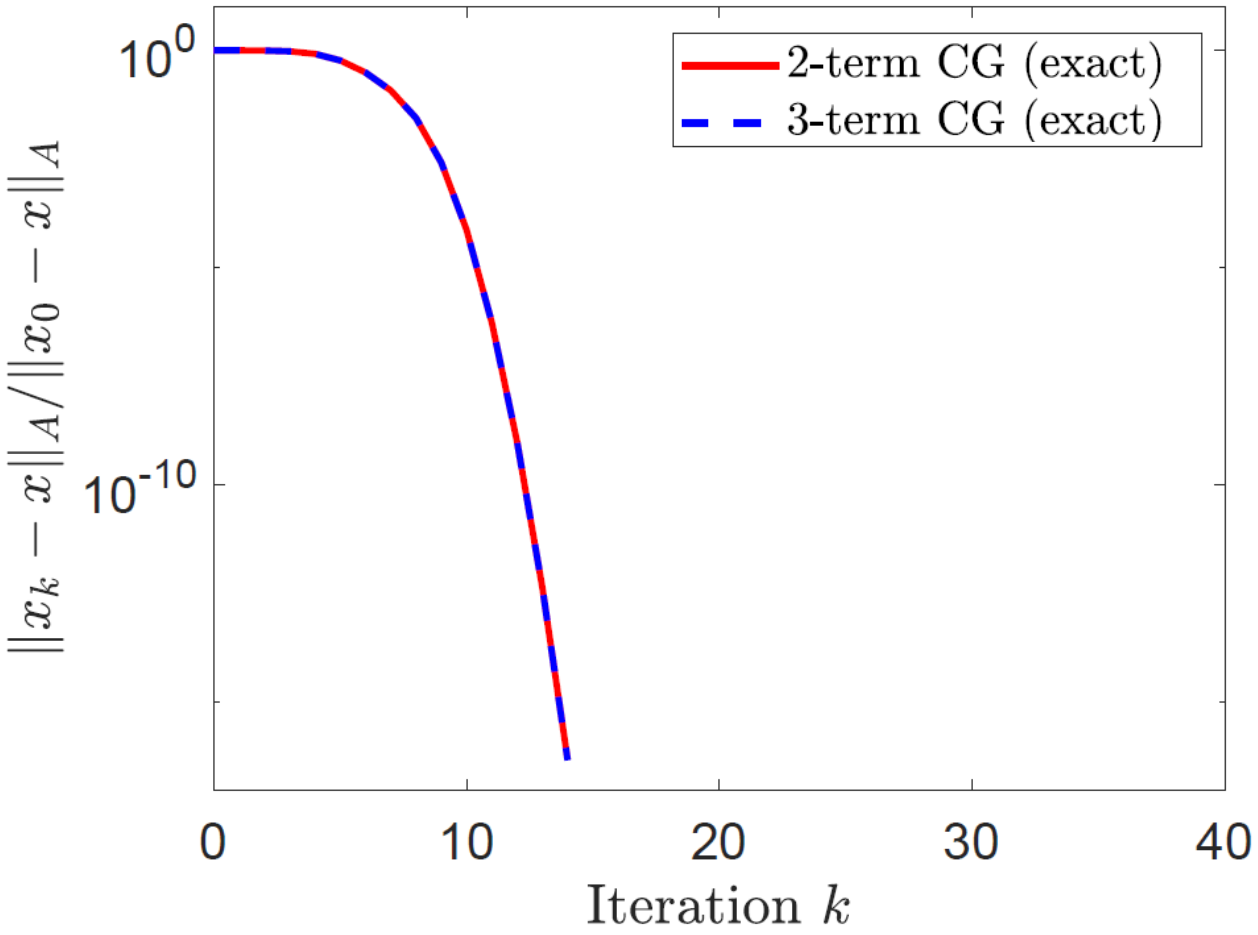
2 algorithmic variants of CG:

1. 2-term recurrence variant of Hestenes and Stiefel
2. 3-term recurrence variant

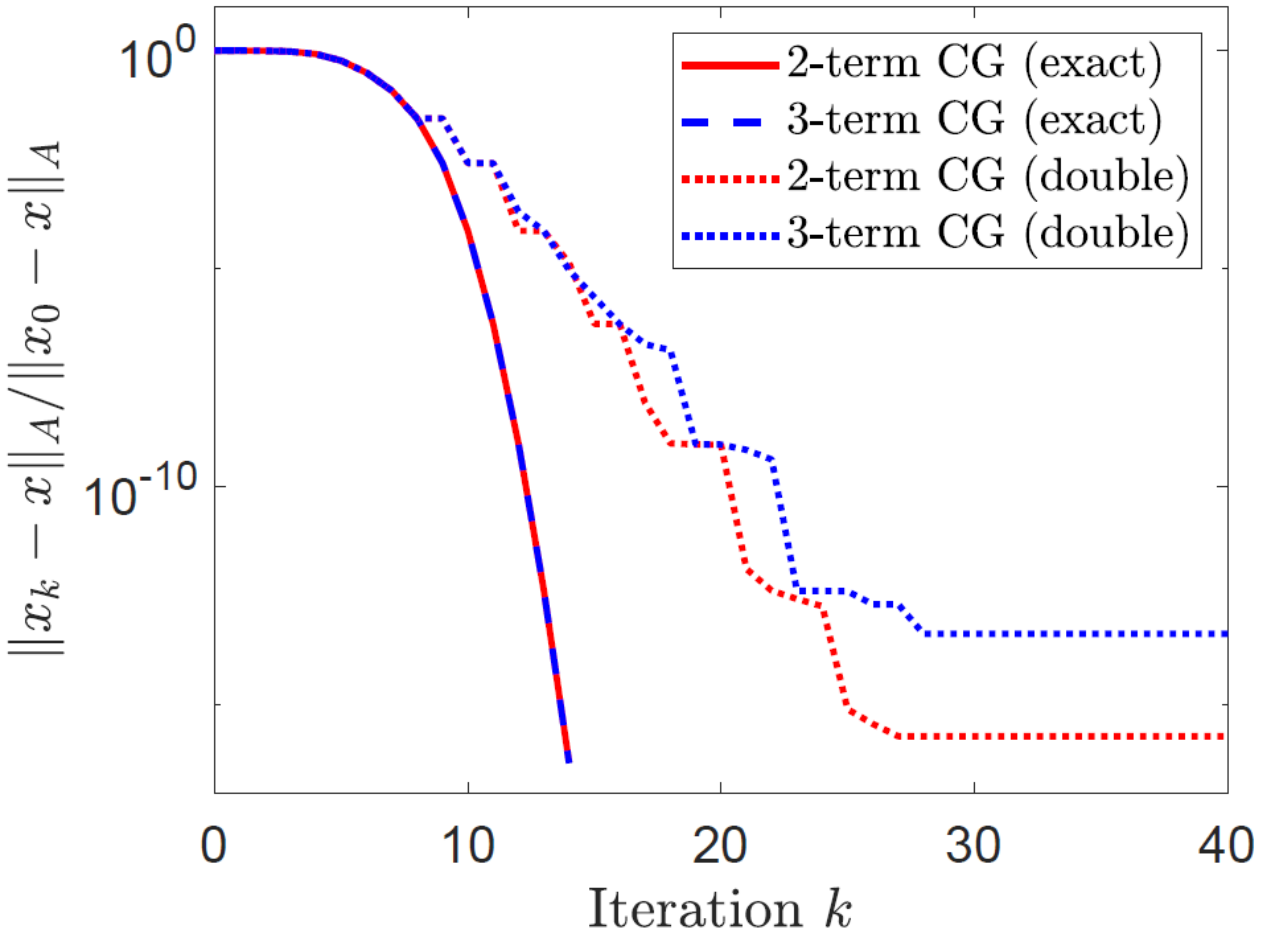
$$x_{k+1} = x_k + \frac{1}{\phi_k}\left(r_k + \psi_{k-1}(x_k - x_{k-1})\right)$$

$$r_{k+1} = r_k + \frac{1}{\phi_k}\left(-Ar_k + \psi_{k-1}(r_k - r_{k-1})\right)$$

Example 7: Computational behavior of different CG algorithms



Example 7: Computational behavior of different CG algorithms



Example 7: Computational behavior of different CG algorithms

- In 2-term recurrence variant, loss of accuracy is caused by a simple **accumulation** of local rounding errors

[Greenbaum, SIMAX 18, 1997]

Example 7: Computational behavior of different CG algorithms

- In 2-term recurrence variant, loss of accuracy is caused by a simple **accumulation** of local rounding errors

[Greenbaum, SIMAX 18, 1997]

- In 3-term recurrence variant, the loss of accuracy is caused by accumulation of local rounding errors, which is then **amplified** (by a factor related to the sizes of adjacent residual norms)

[Gutknecht and Strakoš, SIMAX 22, 2000]

Example 7: Computational behavior of different CG algorithms

- In 2-term recurrence variant, loss of accuracy is caused by a simple **accumulation** of local rounding errors

[Greenbaum, SIMAX 18, 1997]

- In 3-term recurrence variant, the loss of accuracy is caused by accumulation of local rounding errors, which is then **amplified** (by a factor related to the sizes of adjacent residual norms)

[Gutknecht and Strakoš, SIMAX 22, 2000]

- The amplification of local rounding errors is even worse in other algorithmic variants, like pipelined and s-step CG

[C., Rozložník, Strakoš, Tichý, Tůma, SISC 40, 2018], [C., PhD Thesis, 2015]

Example 7: Computational behavior of different CG algorithms

Main point: Rounding errors cause convergence delay and loss of attainable accuracy. These effects depend on the particular algorithm/implementation of CG.

Example 8: Residual versus error and stopping criteria

Setup: Construction of Meurant [Meurant, Num. Algs. 84, 2020]

Let $N = 20$.

Define two sequences of residual 2-norms and error A-norms:

1. $\|r_k\|_2 = 1$ if k is odd, $\|r_k\|_2 = 2$ if k is even for $k = 0, \ldots, N-1$
   $\|e_0\|_A = 1$, $\|e_k\|_A = 0.4\|e_{k-1}\|_A$ for $k = 1, \ldots, N-1$

2. $\|r_0\|_2 = 1$, $\|r_k\|_2 = 0.4\|r_{k-1}\|_2$ for $k = 1, \ldots, N-1$
   $\|e_0\|_A = 1, \|e_k\|_A = 0.999\|e_{k-1}\|_A$ for $k = 1, \ldots, N-1$

Example 8: Residual versus error and stopping criteria

Setup: Construction of Meurant [Meurant, Num. Algs. 84, 2020]

Let $N = 20$.

Define two sequences of residual 2-norms and error A-norms:

1. $\|r_k\|_2 = 1$ if k is odd, $\|r_k\|_2 = 2$ if k is even for $k = 0, \ldots, N-1$
   $\|e_0\|_A = 1$, $\|e_k\|_A = 0.4\|e_{k-1}\|_A$ for $k = 1, \ldots, N-1$

2. $\|r_0\|_2 = 1$, $\|r_k\|_2 = 0.4\|r_{k-1}\|_2$ for $k = 1, \ldots, N-1$
   $\|e_0\|_A = 1, \|e_k\|_A = 0.999\|e_{k-1}\|_A$ for $k = 1, \ldots, N-1$

$\nu_k = 1/\|r_k\|_2 \,, k = 1, \ldots, N-1$

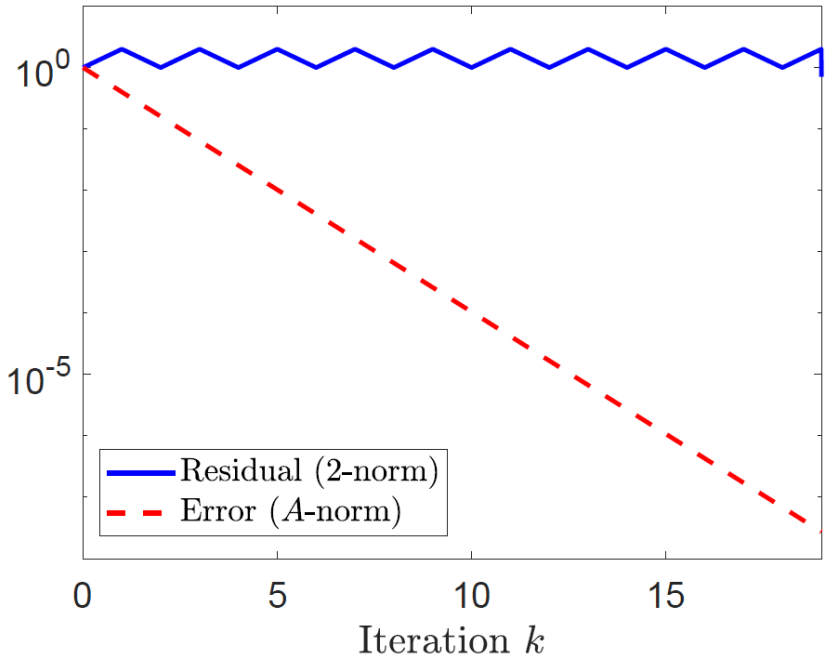
$\sigma_k = \|e_k\|_A^2(\|r_k\|_2\|r_0\|_2), k = 0, \ldots, N-1$

$$L = \begin{bmatrix} \sigma_0 & & & \\ \sigma_1 & \sigma_1\nu_1 & & \\ \vdots & \vdots & \ddots & \\ \sigma_{N-1} & \sigma_{N-1}\nu_1 & \cdots & \sigma_{N-1}\nu_{N-1} \end{bmatrix}$$

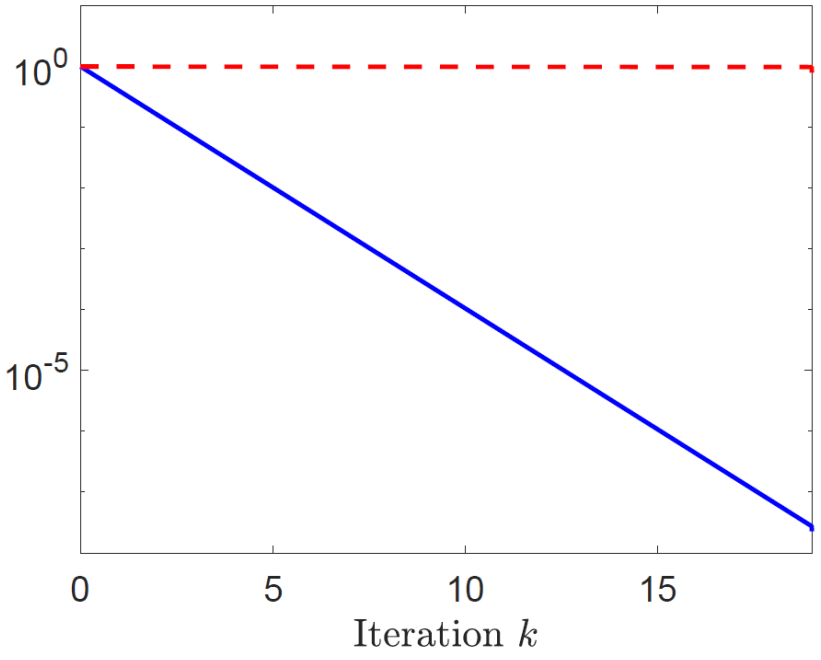
Run exact CG on $A = \left(L + \hat{L}^T\right)^{-1}$, $b = e_1$ (using $x_0 = 0$)

Example 8: Residual versus error and stopping criteria

$\|r_k\|_2 = 1$ if k odd, $\|r_k\|_2 = 2$ if k even for $k = 0, \ldots, N-1$
$\|e_0\|_A = 1$, $\|e_k\|_A = 0.4\|e_{k-1}\|_A$ for $k = 1, \ldots, N-1$



$\|r_0\|_2 = 1$, $\|r_k\|_2 = 0.4\|r_{k-1}\|_2$ for $k = 1, \ldots, N-1$
$\|e_0\|_A = 1$, $\|e_k\|_A = 0.999\|e_{k-1}\|_A$ for $k = 1, \ldots, N-1$



Example 8: Residual versus error and stopping criteria

- Hestenes and Stiefel comment that for *any* prescribed sequence of residual 2-norms, there exists an SPD A and RHS b such that CG exhibits the prescribed convergence behavior (see Section 18)

Theorem 18:3. *There is no restriction whatever on the positive constants* a_i, b_i *in the cg-process, that is, given two sequences of positive numbers* $a_0, a_1, \ldots, a_{n-1}$ *and* $b_0, b_1, \ldots, b_{n-1}$, *there is a symmetric positive definite matrix* A *and a vector* r_0 *such that the cg-algorithm applied to* A, r_0 *yield the given numbers.*

Example 8: Residual versus error and stopping criteria

- Hestenes and Stiefel comment that for *any* prescribed sequence of residual 2-norms, there exists an SPD A and RHS b such that CG exhibits the prescribed convergence behavior (see Section 18)

> *Theorem* 18:3. *There is no restriction whatever on the positive constants* a_i, b_i *in the cg-process, that is, given two sequences of positive numbers* $a_0, a_1, \ldots, a_{n-1}$ *and* $b_0, b_1, \ldots, b_{n-1}$, *there is a symmetric positive definite matrix* A *and a vector* r_0 *such that the cg-algorithm applied to* A, r_0 *yield the given numbers.*

- This means that small residual $\not\to$ small error, large residual $\not\to$ large error
- Thus residual norm is not a reliable stopping criterion (but it is often used, since it is computable and cheap)
- Lots of work on error estimation and stopping criteria for CG

[Hestenes and Stiefel, 1952 (Section 4)], [Golub and Meurant, BIT 37, 1997], [Golub and Strakoš Num. Algs. 8, 1994], [Meurant, Papež, and Tichý, Num. Algs. 88, 2021], [Strakoš and Tichý, ETNA 13, 2002], [Strakoš and Tichý, BIT 45, 2005],

Example 8: Residual versus error and stopping criteria

Main point: The residual 2-norm is not a reliable indicator of the error in CG.

Example 9: CG with "average" initial residuals

Setup:

Matrix A and particular RHS b_* come from discretization of a BVP with Dirichlet boundary conditions in [Morin, Nochetto, and Siebert, SIREV 44, 2002].

With standard FE discretization using standard uniform triangulation, we get a system with $N = 3969$

Two preconditioners:

1. Laplace operator preconditioning
2. Algebraic incomplete Cholesky factorization of A with drop tolerance 10^{-2}

Example 9: CG with "average" initial residuals

Setup:

Matrix A and particular RHS b_* come from discretization of a BVP with Dirichlet boundary conditions in [Morin, Nochetto, and Siebert, SIREV 44, 2002].

With standard FE discretization using standard uniform triangulation, we get a system with $N = 3969$

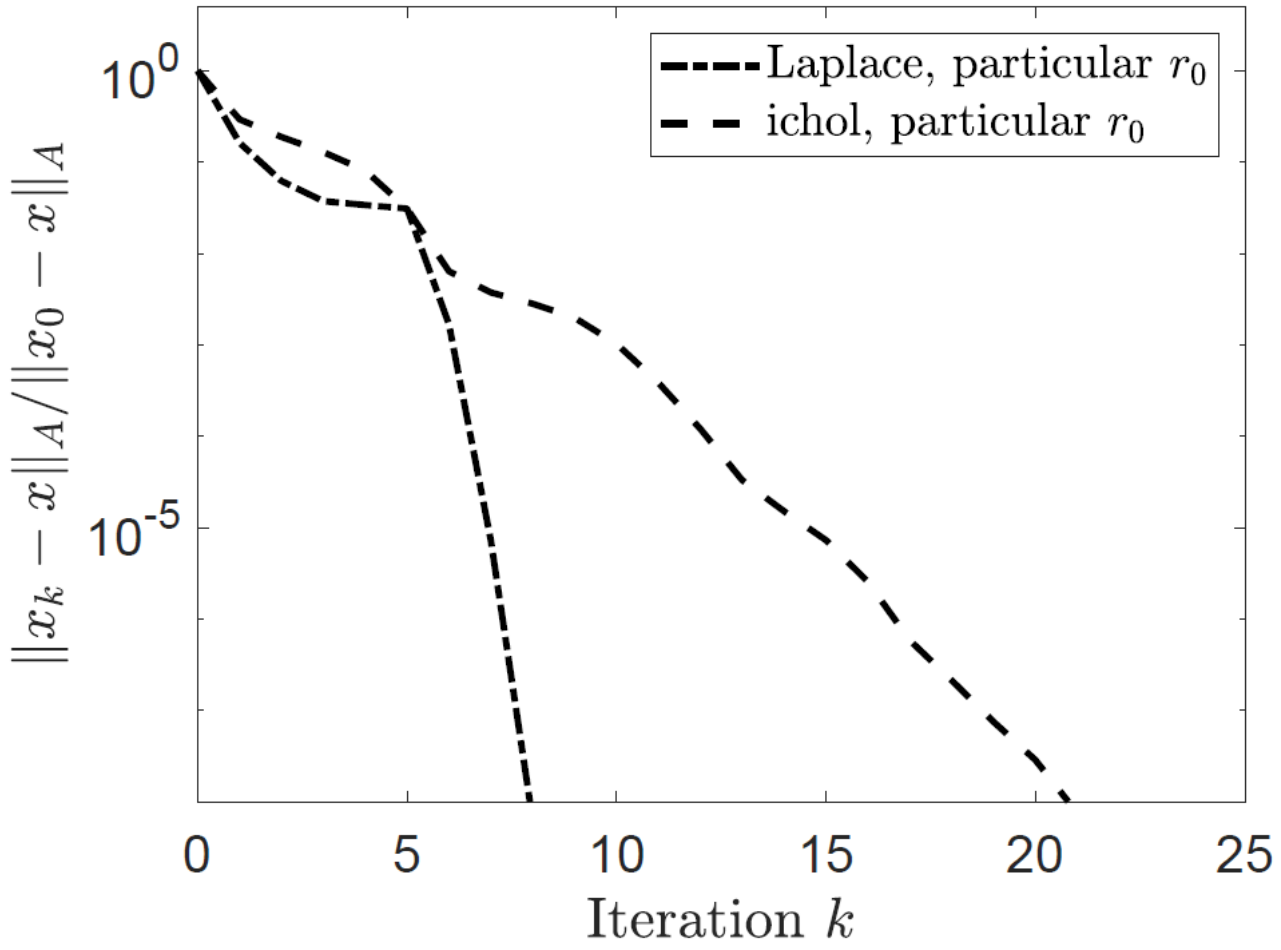
Two preconditioners:

1. Laplace operator preconditioning
2. Algebraic incomplete Cholesky factorization of A with drop tolerance 10^{-2}

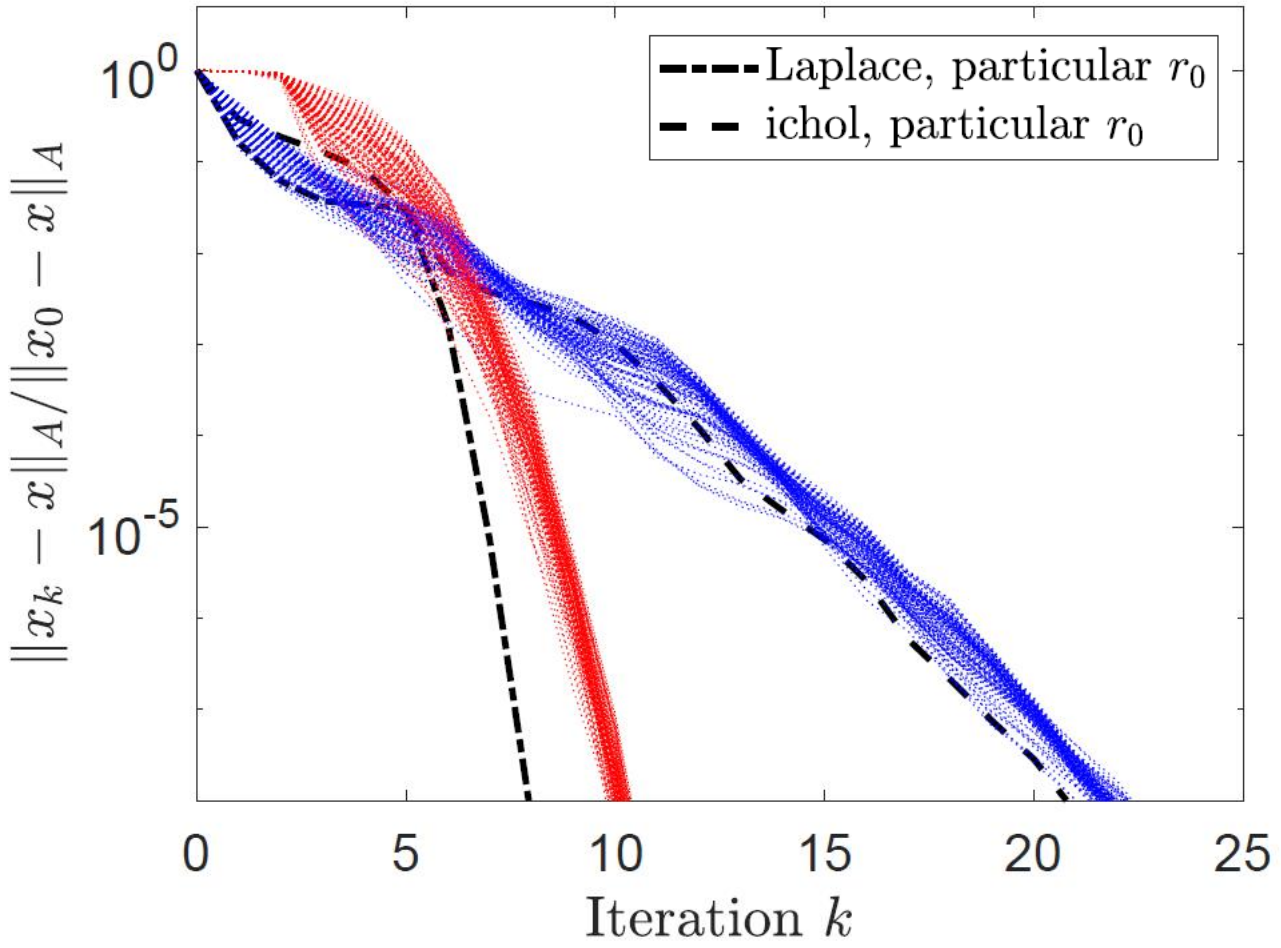
Run CG in double precision on preconditioned systems with A and normalized b_*, and also 100 normalized random RHS's generated via `randn(N,1)`

- For $x_0 = 0$, this results in random initial residuals r_0

Example 9: CG with "average" initial residuals



Example 9: CG with "average" initial residuals



Example 9: CG with "average" initial residuals

$$\|x - x_k\|_A = \min_{p \in P_k(0)} \|p(A)(x - x_0)\|_A = \min_{p \in P_k(0)} \left(\sum_{i=1}^{N} \eta_i^2 \frac{p(\lambda_i)^2}{\lambda_i} \right)^{1/2}$$

- It might seem that an "average" or "random" initial residual will give enough information about CG on real problems, but this is not the case

[Gergelits, Mardal, Nielsen, and Strakoš, SINUM 57, 2019]

Example 9: CG with "average" initial residuals

Main point: The terms "average" and "random" have very specific meanings and should be used carefully. "Average" and "random" cases can not give a good indication of CG behavior for more general problems.

Example 10: The trajectory of finite precision CG computations

Setup:

A: Diagonal matrix with $N = 35, \lambda_1 = 0.1, \lambda_N = 10^2, \rho = 0.65$

$b = [1, \ldots, 1]^T / \sqrt{N}\,, x_0 = 0$

Run CG in double to produce iterates $\bar{x}_k$ and exact CG to produce iterates x_k

Example 10: The trajectory of finite precision CG computations

Setup:

A: Diagonal matrix with $N = 35, \lambda_1 = 0.1, \lambda_N = 10^2, \rho = 0.65$

$b = [1, \ldots, 1]^T/\sqrt{N}\,, x_0 = 0$

Run CG in double to produce iterates $\bar{x}_k$ and exact CG to produce iterates x_k

Shift double precision iterates according to sequence

$$\ell(k) = \max\{\, i \mid \operatorname{rank}(\mathcal{K}_i(A, r_0) = k) \,\}, \qquad k = 1,2, \ldots$$

Example 10: The trajectory of finite precision CG computations

Setup:

A: Diagonal matrix with $N = 35, \lambda_1 = 0.1, \lambda_N = 10^2, \rho = 0.65$

$b = [1, \ldots, 1]^T/\sqrt{N}\,, x_0 = 0$

Run CG in double to produce iterates $\bar{x}_k$ and exact CG to produce iterates x_k

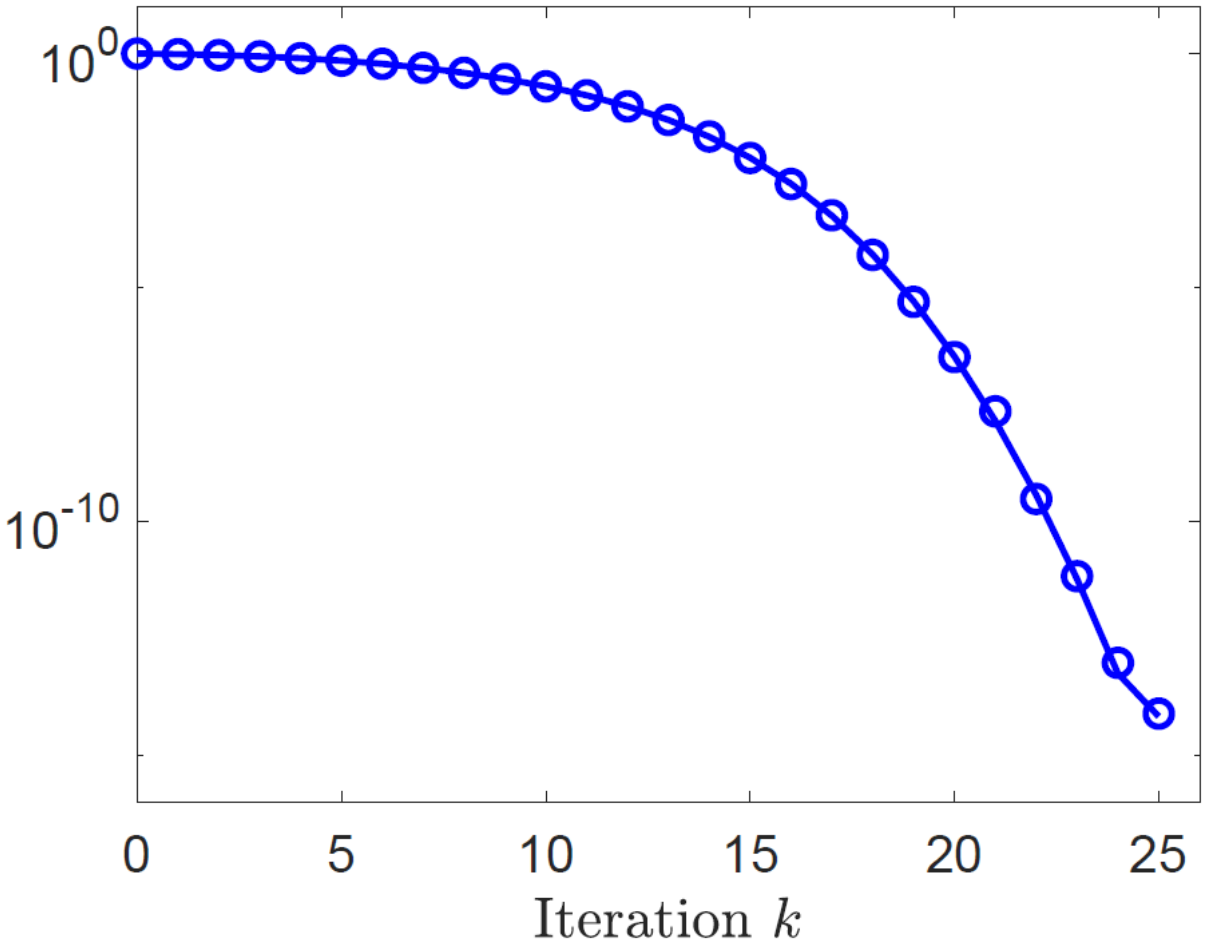
Shift double precision iterates according to sequence

$$\ell(k) = \max\{\, i \mid \operatorname{rank}(\mathcal{K}_i(A, r_0) = k) \,\}, \qquad k = 1,2, \ldots$$

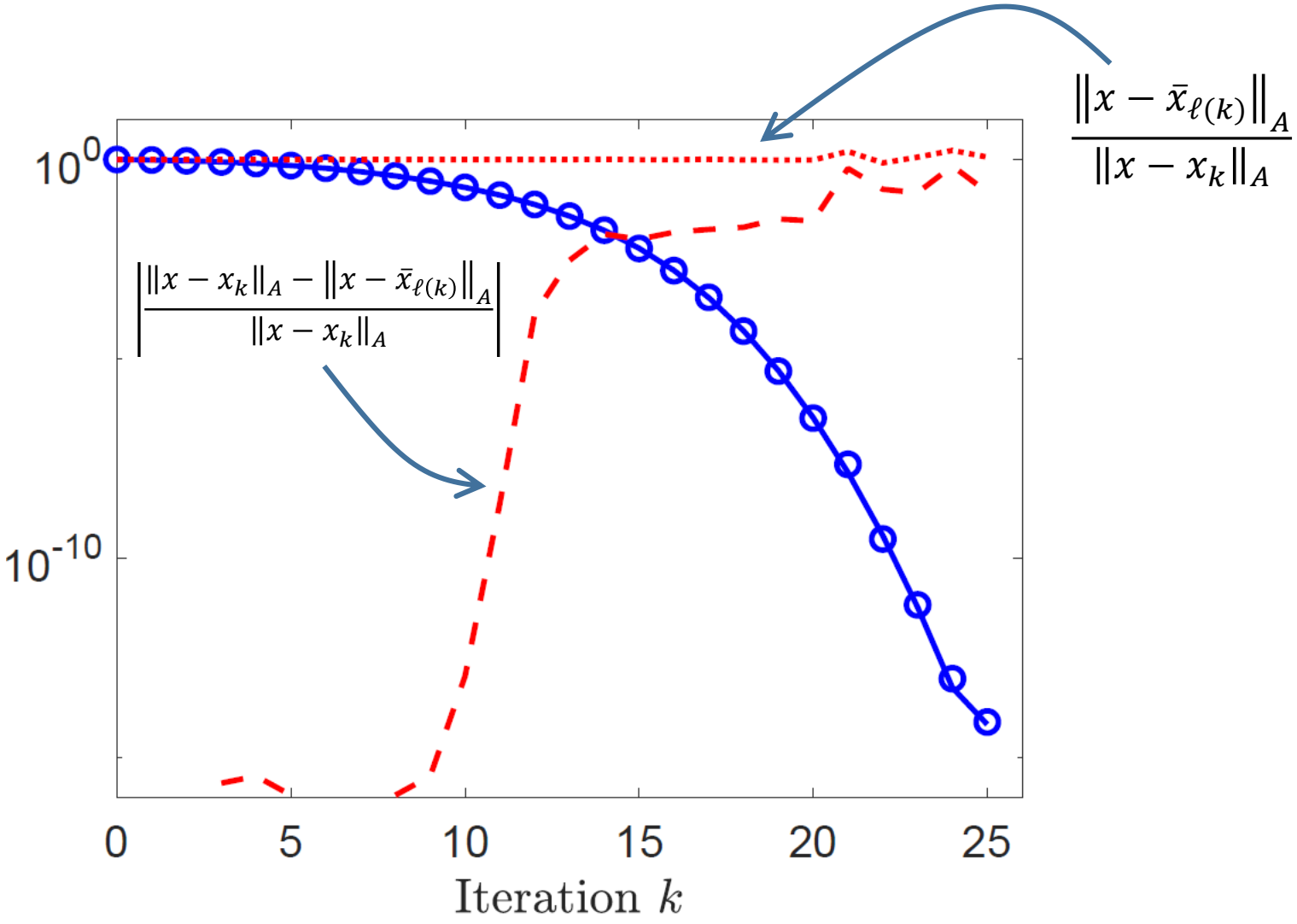
Quantities of interest:

$$\frac{\|x - \bar{x}_{\ell(k)}\|_A}{\|x - x_k\|_A}, \qquad \left| \frac{\|x - x_k\|_A - \|x - \bar{x}_{\ell(k)}\|_A}{\|x - x_k\|_A} \right|$$

Example 10: The trajectory of finite precision CG computations



Example 10: The trajectory of finite precision CG computations



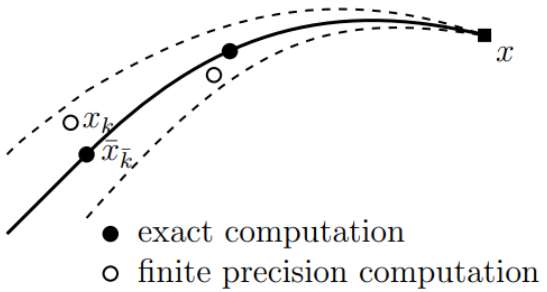
Example 10: The trajectory of finite precision CG computations

$\frac{\|x-\bar{x}_{\ell(k)}\|_A}{\|x-x_k\|_A} \approx 1$ means that the convergence trajectories for finite precision CG and exact CG are close to being identical.

Example 10: The trajectory of finite precision CG computations

$\frac{\|x-\bar{x}_{\ell(k)}\|_A}{\|x-x_k\|_A} \approx 1$ means that the convergence trajectories for finite precision CG and exact CG are close to being identical.

If $\left|\frac{\|x-x_k\|_A-\|x-\bar{x}_{\ell(k)}\|_A}{\|x-x_k\|_A}\right| < 1$, this means we can consider the trajectory of finite precision CG iterates to be enclosed within a narrow tunnel around the trajectory of exact CG iterates



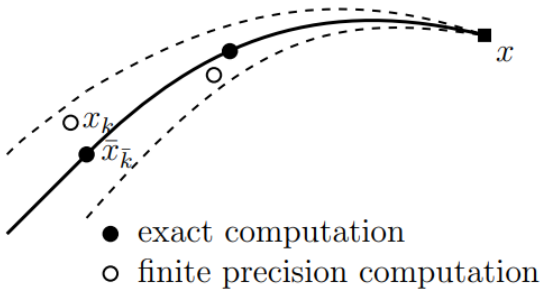
[Gergelits, Master's Thesis, 2013]

Example 10: The trajectory of finite precision CG computations

$\frac{\|x-\bar{x}_{\ell(k)}\|_A}{\|x-x_k\|_A} \approx 1$ means that the convergence trajectories for finite precision CG and exact CG are close to being identical.

If $\left|\frac{\|x-x_k\|_A-\|x-\bar{x}_{\ell(k)}\|_A}{\|x-x_k\|_A}\right| < 1$, this means we can consider the trajectory of finite precision CG iterates to be enclosed within a narrow tunnel around the trajectory of exact CG iterates

These appear to hold for this example, but have not been proven to hold in general!



[Gergelits, Master's Thesis, 2013]

[Gergelits, Master's Thesis, 2013], [Gergelits, Hnětynková, Kubínová, Proc. HPCSE 2017]

Example 10: The trajectory of finite precision CG computations

Main point:
The approximate solutions produced by finite precision CG can be mapped to those produced by exact CG via a mapping defined by the rank deficiency of the Krylov subspace basis.

It seems that the trajectory of the approximate solutions remains in a narrow "tunnel" around those produced by exact CG.

Summary

- Krylov subspace methods are remarkable mathematical objects
- 70 years of investigation
- But we still have things to discover!

C., Jörg Liesen, and Zdeněk Strakoš. "70 years of Krylov subspace methods: The journey continues." *arXiv preprint arXiv:2211.00953* (2022).

Summary

- Krylov subspace methods are remarkable mathematical objects
- 70 years of investigation
- But we still have things to discover!

C., Jörg Liesen, and Zdeněk Strakoš. "70 years of Krylov subspace methods: The journey continues." *arXiv preprint arXiv:2211.00953* (2022).

Thank You!

carson@karlin.mff.cuni.cz
www.karlin.mff.cuni.cz/~carson/