MA3227 Numerical Analysis II

Lecture 9: Krylov Methods Applied to Poisson's Equation

Simon Etter

2019/2020

Krylov Methods Applied to Poisson's Equation

Introduction

Our overarching goal in the first half of this module is to develop solvers for the finite-difference-discretised Poisson equation $-\Delta_n^{(d)} u_n = f$ which achieve optimal $\mathcal{O}(N) = \mathcal{O}(n^d)$ runtime and memory.

We have seen that LU factorisation achieves this goal only for $d = 1$.

This lecture addresses the following question:

Can Krylov methods achieve $\mathcal{O}(N)$ runtime and memory for any d?

Krylov Methods Applied to Poisson's Equation

Preparing for the discussion

- $-\Delta_n^{(d)}$ is spd; hence we can use the conjugate gradient algorithm.
- Runtime of conjugate gradients is $\mathcal{O}(Nk)$. Memory is $\mathcal{O}(N)$.
- The eigenvalues of $-\Delta_n^{(d)}$ are asymptotically contained in

$$\mathcal{E} = \left[d\,\pi^2, 4\,d\,(n+1)^2\right].$$

We have seen in Lecture 2 that the eigenvalues of $\Delta_n^{(1)}$ are given by

$$\lambda_k = (n+1)^2\left(2\cos\left(\pi\tfrac{k}{n+1}\right) - 2\right) \in [-4\,(n+1)^2, -\pi^2 + \mathcal{O}(n^{-2})].$$

$\mathcal{E}$ for $\Delta_n^{(d)}$ follows from the Kronecker-product argument on page 8 of Lecture 3.

- If all eigenvalues of A are contained in $[1, \kappa]$, the conjugate gradient iterates x_k satisfy the bound

$$\|Ax_k - b\| \le C\,\rho^k \qquad \text{where} \qquad \rho = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}.$$

- We are usually interested in achieving a fixed error $\|Ax_k - b\| \le \varepsilon$; hence we should choose $k = \frac{\log(\varepsilon/C)}{\log(\rho)}$

Krylov Methods Applied to Poisson's Equation

Putting things together

Combining the last three points yields

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{4\,(n+1)^2}{\pi^2} = \mathcal{O}(n^2),$$
$$\rho = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} = \frac{1-\kappa^{-1/2}}{1+\kappa^{-1/2}} = 1-\mathcal{O}(\kappa^{-1/2}) = 1-\mathcal{O}(n^{-1}),$$
$$k = \frac{\log(\varepsilon/C)}{\log(\rho)} = \mathcal{O}\Big(\log(\varepsilon)\,(1-\rho)^{-1}\Big) = \mathcal{O}\Big(\log(\varepsilon)\,n\Big).$$

Conclusion

To achieve a fixed error ε independent of n, we have to choose $k \propto n$.

Thus, the runtime of conjugate gradients is

$$\mathcal{O}(Nk) = \mathcal{O}(n^{d+1}) = \mathcal{O}(N^{(d+1)/d}).$$

This is not the desired $\mathcal{O}(N)$ scaling. How does it compare to LU?

Krylov Methods Applied to Poisson's Equation

Discussion: LU vs. conjugate gradients

Conjugate gradients has optimal $\mathcal{O}(N)$ memory; hence it is as good as LU for $d = 1$, and better than LU for $d > 1$.

The situation is more complicated for the runtime:

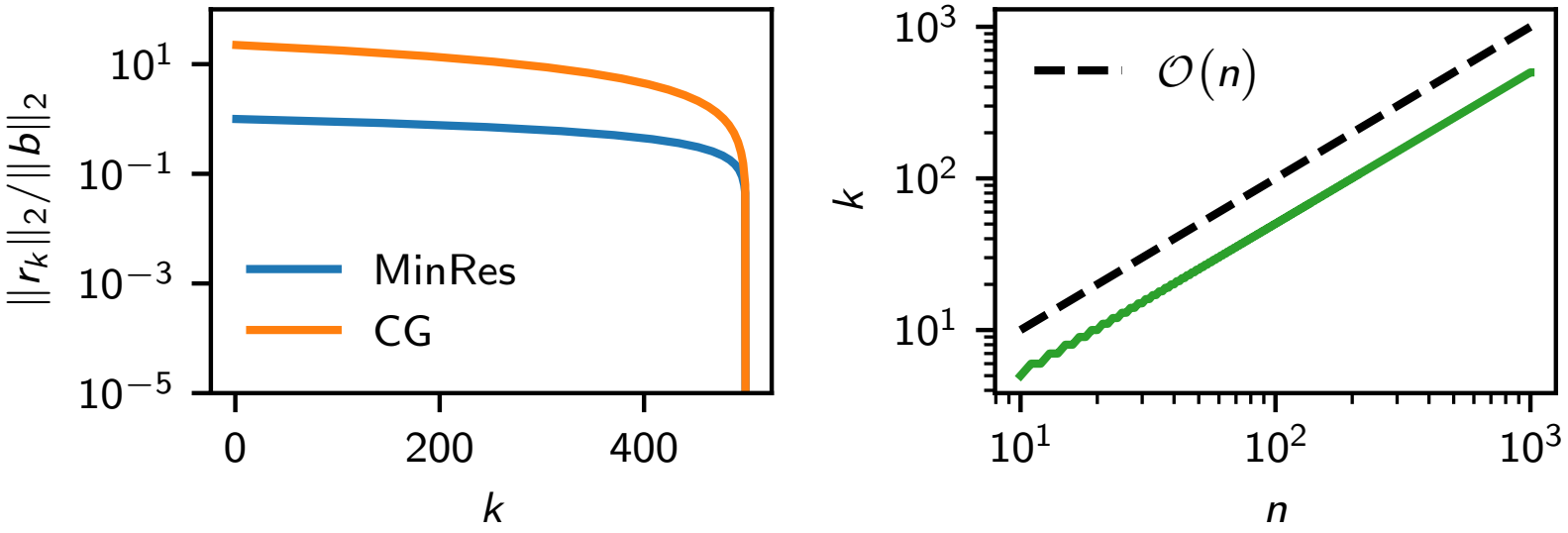
	LU runtime	CG runtime
$d = 1$	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$
$d = 2$	$\mathcal{O}(N^{3/2})$	$\mathcal{O}(N^{3/2})$
$d = 3$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^{4/3})$

Outlook

The following slides demonstrate the convergence of Krylov methods when applied to the Poisson equation.

Krylov Methods Applied to Poisson's Equation

One-dimensional Poisson equation ($n = 1000$)

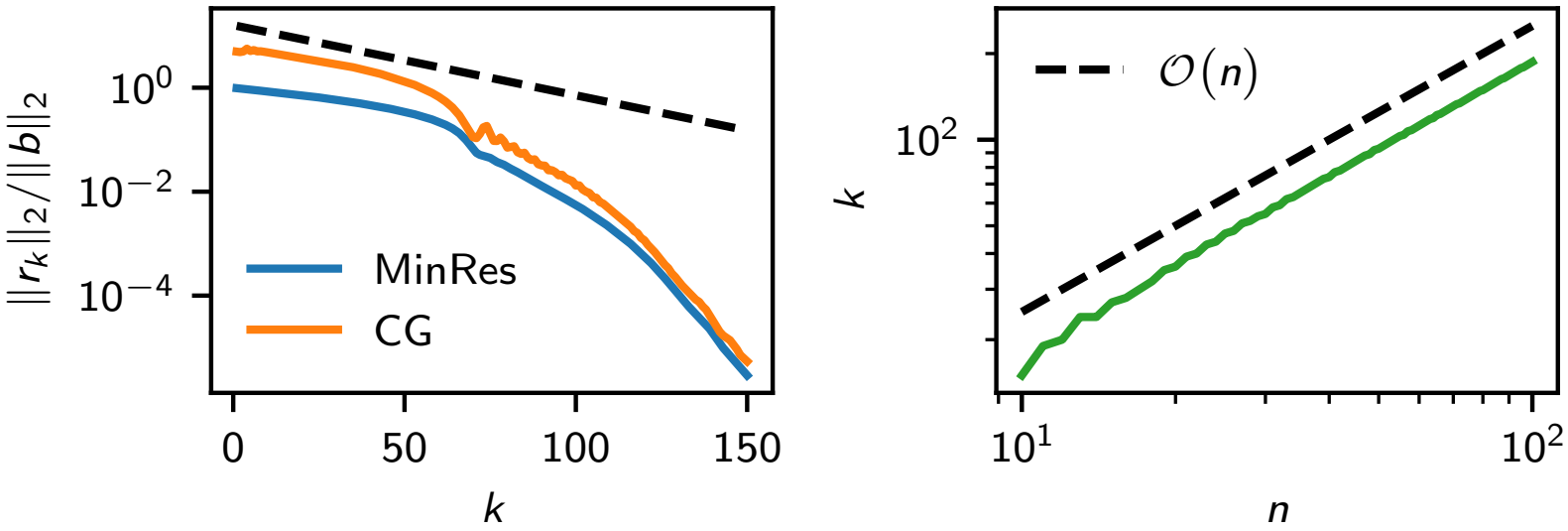


Left: convergence history of MinRes and conjugate gradients for $n = 1000$.
Right: number of CG iterations to achieve $\|Ax_k - b\|_2 \leq 10^{-8} \|b\|_2$.

As predicted, we get hardly any convergence for $k \ll n$ (left plot), and the number of iterations k required to achieve a fixed error is $\mathcal{O}(N)$ (right plot).

Krylov Methods Applied to Poisson's Equation

Two-dimensional Poisson equation



Left: convergence history of MinRes and conjugate gradients for $n = 100$.
Right: number of CG iterations to achieve $\|Ax_k - b\|_2 \leq 10^{-8} \|b\|_2$.

Again, convergence kicks in only for $k \approx n$ (left plot), and we observe $k = \mathcal{O}(n)$ (right plot).

Additionally, we observe that $\|Ax_k - b\|_2$ does not decay monotonically for conjugate gradients. This is expected since conjugate gradients minimises $\|x_k - x\|_A$ rather than $\|Ax_k - b\|_2$.

Krylov Methods Applied to Poisson's Equation

Discussion

The above seems to suggest that Krylov methods are not very helpful.

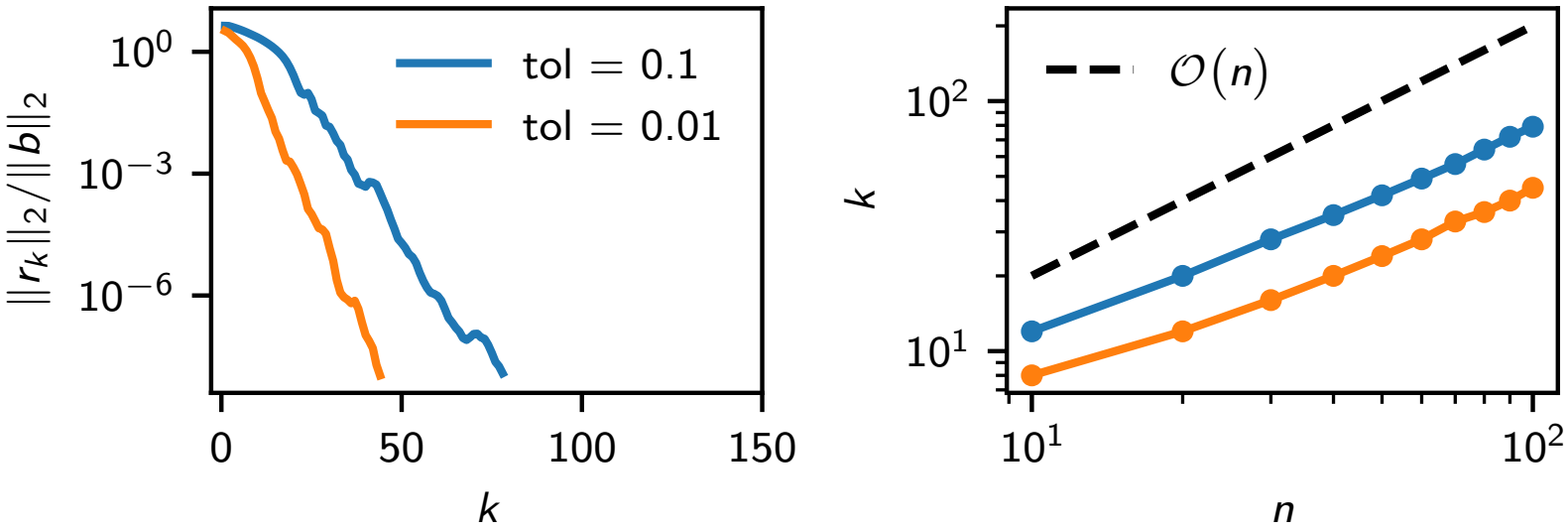
This is partly because so far we have not made use of preconditioning: if $Ax = b$ is too hard to solve, then solve $P^{-1}Ax = P^{-1}b$ instead where P is some matrix which makes the problem easier.

The next slide demonstrates the effect of the ILU(τ) preconditioner introduced in Lecture 7. This preconditioner is given by $P = \tilde{L}\tilde{U}$ where $\tilde{L}, \tilde{U}$ are computed by applying LU factorisation to A but ignoring entries which are smaller than $\texttt{tol} \times 4\,(n+1)^2$.

$4\,(n+1)^2$ is the magnitude of the largest entry in $\Delta_n^{(2)}$.

Krylov Methods Applied to Poisson's Equation

Two-dimensional Poisson equation with ILU preconditioning



Left: convergence history of conjugate gradients for $n = 100$.
Right: number of CG iterations to achieve $\|Ax_k - b\|_2 \leq 10^{-8} \|b\|_2$.

Unsurprisingly, the convergence improves for smaller `tol`. However, smaller `tol` also means that the preconditioner becomes more expensive to compute and apply.

Overall, we conclude that ILU preconditioning reduces the prefactors but does not change the $k = \mathcal{O}(n)$ scaling.

Krylov Methods Applied to Poisson's Equation

Summary / Discussion

- Krylov methods almost always require a preconditioner to be effective.
- ILU preconditioning is a simple "black-box" preconditioner which may help reduce the prefactors, but typically it does not lead to asymptotically better runtimes.