R. Herzog, G. Müller, E. Herberg, M. Hashemi
Heidelberg University

Exercise 3 - Solution

Date issued: 2nd May 2023
Date due: 9th May 2023

Homework Problem 3.1 (Comparison of SD and CG Convergence) 5 Points

(*i*) Explain the differences and (dis-)advantages of the worst-case convergence results of the steepest descent and CG method for solving s. p. d. linear systems.

(*ii*) Plot the maximum number of iterations needed by the steepest descent and the CG method, respectively, (as guaranteed by their worst case convergence estimates), to produce a reduction of 10^{-6} in the error (relative to the initial error).

Solution.

(*i*) For the iterates $\widetilde{x}^{(k)}$ and $x^{(k)}$ of the steepest descent and the CG method, respectively, the convergence results are

$$\|\widetilde{x}^{(k+1)} - x^*\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right) \|\widetilde{x}^{(k)} - x^*\|_A \quad \text{for all } k \geq 0, \tag{SD}$$

$$\|x^{(k)} - x^*\|_A \leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k \|x^{(0)} - x^*\|_A, \quad \text{for all } k \geq 0. \tag{CG}$$

Since $\kappa \geq 1$, we know that

$$\frac{\kappa - 1}{\kappa + 1}, \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \in (0, 1),$$

accordingly, the steepest descent method is at least Q-linearly convergent while the CG method is at least R-linearly convergent. Structurally, the Q-linear convergence of the steepest descent method is favourable, as it guarantees a relative decrease in each step and implies R-linear

convergence with the same constant. However, the resulting R-linear convergence bound of the steepest descent method is generally worse than that of the CG method since

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} = \frac{\overbrace{1-\frac{1}{\sqrt{\kappa}}}^{\leq 1-\frac{1}{\kappa}}}{\underbrace{1+\frac{1}{\sqrt{\kappa}}}_{\geq 1+\frac{1}{\kappa}}} \leq \frac{1-\frac{1}{\kappa}}{1+\frac{1}{\kappa}} = \frac{\kappa-1}{\kappa+1}.$$

I. e. the steepest descent generally has to be expected to potentially converge much slower than the CG method. (2 Points)

(*ii*) To get an idea how convenient the better worst case estimate of the CG is in practice, consider an initial starting point $x^{(0)}$ for both the steepest descent and the CG method. We can compute

$$\left(\frac{\kappa-1}{\kappa+1}\right)^k \leq \varepsilon \Leftrightarrow k \geq \left\lceil \frac{\ln(\varepsilon)}{\ln\left(\frac{\kappa-1}{\kappa+1}\right)} \right\rceil \tag{0.2}$$

$$2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \leq \varepsilon \Leftrightarrow k \geq \left\lceil \frac{\ln(\varepsilon/2)}{\ln\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)} \right\rceil \tag{0.3}$$

to obtain a upper bound on the number of iterations needed for the A-norm of the errors $x^{(k)} - x^{(0)}$ and $\widetilde{x}^{(k)} - x^{(0)}$ to be reduced by $\varepsilon \in (0,1)$ relative to the initial error. As expected, the number of iterations needed for varying condition numbers grows much more rapidly in the case of the steepest descent bound. As we know, the results in Algorithm 4.6 and Corollary 4.20 agree very well with the bounds used here and are not plotted additionally.

Note that the CG bound can additionally be capped by the space dimension due to the expanding subspace minimization property. (3 Points)

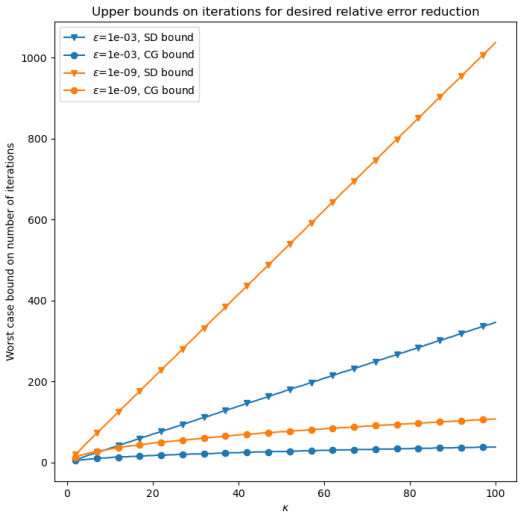

Figure 0.1: Evolution of the upper bounds on the number of iterations for relative decrease in the error.

Homework Problem 3.2 (Any CG-history is better than no history.) 5 Points

Let $A \in \mathbb{R}^{n\times n}$ be s. p. d., $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Further, let $x^{(k)}$ be an iterate of the CG method for minimizing $\phi(x) = \frac{1}{2}x^\mathsf{T}Ax - b^\mathsf{T}x + c$ and let $x^{(k+1)}$ be computed by an additional step of the CG method and $\widetilde{x}^{(k+1)}$ be computed by a steepest descent step with Cauchy stepsize starting from $x^{(k)}$.

Show that $\|x^{(k+1)} - x^*\|_A \leq \|\widetilde{x}^{(k+1)} - x^*\|_A$.

Solution.

For $k = 0$ this is immediately clear.

For any other k, since the CG method is a method of conjugate directions, the iterate $x^{(k+1)}$ minimizes ϕ over the affine subspace $x^{(0)} + \operatorname{span}\{d^{(0)}, d^{(1)}, \ldots, d^{(k)}\}$ (Lemma 4.14 Statement *(ii)*). Accordingly, it also minimizes the function $\phi(x) - \phi(x^*) = \frac{1}{2}\|x - x^*\|_A^2$ and by monotonicity of the square root function it minimizes the error $\|x - x^*\|_A$ over the same space. (2 Points)

Accordingly, we obtain the claim immediately by showing that $\widetilde{x}^{(k+1)} \in x^{(0)} + \operatorname{span}\{d^{(0)}, d^{(1)}, \ldots, d^{(k)}\}$ as well. There are at least two options this.

The easiest way is to recognize that since $d^{(k)} = -M^{-1}r^{(k)} + \beta^{(k)}d^{(k-1)}$ we have that

$$\begin{aligned}
\widetilde{x}^{(k+1)} &= x^{(k)} + \alpha^{(k)}\left(-\nabla_M\phi(x^{(k)})\right)(-\nabla_M\phi(x^{(k)})) \\
&= x^{(k)} + \alpha^{(k)}\left(-M^{-1}r^{(k)}\right)(-M^{-1}r^{(k)}) \\
&= \underbrace{x^{(k)}}_{\in x^{(0)}+\operatorname{span}\{d^{(0)},d^{(1)},\ldots,d^{(k-1)}\}} + \underbrace{\alpha^{(k)}\left(-M^{-1}r^{(k)}\right)(d^{(k)} - \beta^{(k)}d^{(k-1)})}_{\in \operatorname{span}\{d^{(k-1)}\,d^{(k)}\}} \\
&\in x^{(0)} + \operatorname{span}\{d^{(0)}, d^{(1)}, \ldots, d^{(k)}\}.
\end{aligned}$$

Alternatively, according to Equation (4.26) in Lemma 4.16 of the script, we know that

$$\begin{aligned}
x^{(0)} + \operatorname{span}\{d^{(0)}, d^{(1)}, \ldots, d^{(k)}\} &= x^{(0)} + M^{-1}\mathcal{K}^{(k+1)}(A\,M^{-1}; r^{(0)}) \\
&= x^{(0)} + M^{-1}\operatorname{span}\{r^{(0)}, (A\,M^{-1})\,r^{(0)}, \ldots, (A\,M^{-1})^k\,r^{(0)}\} \\
&= x^{(0)} + M^{-1}\operatorname{span}\{r^{(0)}, r^{(1)}, \ldots, r^{(k)}\}
\end{aligned}$$

(1 Point)

Additionally,

$$\begin{aligned}
\widetilde{x}^{(k+1)} = x^{(k)} - \alpha^{(k)}\nabla_M\phi(x^{(k)}) &= \underbrace{x^{(k)}}_{\in x^{(0)}+M^{-1}\operatorname{span}\{r^{(0)},r^{(1)},\ldots,r^{(k-1)}\}} - \alpha^{(k)}M^{-1}(r^{(k)}) \\
&\in x^{(0)} + M^{-1}\operatorname{span}\{r^{(0)}, r^{(1)}, \ldots, r^{(k)}\} \\
&= x^{(0)} + \operatorname{span}\{d^{(0)}, d^{(1)}, \ldots, d^{(k)}\}.
\end{aligned}$$

Since the new position after the gradient step is included in the space the CG iterate minimizes the objective and hence the error over, we have the claim. (2 Points)

Homework Problem 3.3 (Clustering of Eigenvalues) 6 Points

Let $A, M \in \mathbb{R}^{n\times n}$ be s. p. d., $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Show that the M-preconditioned CG method will find the unique minimizer of $\phi(x) = \frac{1}{2}x^\mathsf{T}Ax - b^\mathsf{T}x + c$ in $k \leq n$ iterations or less if any of the following conditions is satisfied.

(i) A has only k distinct generalized eigenvalues w.r.t. M;

(ii) The initial error $x^{(0)} - x^*$ is in the span of k distinct generalized eigenvectors of A w.r.t. M;

(*iii*) The initial M-gradient $M^{-1}r^{(0)}$ is in the span of k distinct generalized eigenvectors of A w.r.t. M.

Solution.

The basis of these results is the equality

$$\|e^{(k)}\|_A = \min\left\{\|p(M^{-1}A)\, e^{(0)}\|_A \,\middle|\, p \in \Pi_k,\ p(0) = 1\right\} \tag{4.31}$$

of the lecture notes, which states that we can find the norm of the A-error of the k-th iterate by minimizing the expression $\|p(M^{-1}A)\, e^{(0)}\|_A$ over a certain set of polynomials. This lead to the estimate

$$\|e^{(k)}\|_A \leq \min\left\{\max_{j=1,\dots,n} |p(\lambda^{(j)})|\, \|e^{(0)}\|_A \,\middle|\, p \in \Pi_k,\ p(0) = 1\right\}. \tag{o.4}$$

(*i*) When A only has k distinct generalized eigenvalues $\lambda^{(i^{(1)})}, \dots, \lambda^{(i^{(k)})} > 0$, then the unique interpolating polynomial of maximum degree k for the $k+1$ points $\{(\lambda^{(i^{(j)})}, 0) \mid j = 1, \dots, k\} \cup \{(0,1)\}$ vanishes in all generalized eigenvalues $\lambda^{(1)}, \dots, \lambda^{(n)}$ of A and therefore (o.4) implies that the error vanishes at iteration k at the latest. Due to the simple structure of the interpolation problem, we can explicitly define this polynomial, which ends up being

$$p(x) := \frac{(-1)^k}{\prod_{j=1}^k \lambda^{(i^{(j)})}} \prod_{j=1}^k (x - \lambda^{(i^{(j)})}).$$

(2 Points)

(*ii*) When

$$x^{(0)} - x^* = \sum_{j=1}^k \gamma^{(j)} v^{(i^{(j)})}$$

for generalized eigenvectors $v^{(i^{(j)})}$ and coefficients $\gamma^{(j)}$ then Equation (4.31) yields

$$\begin{aligned}
\|e^{(k)}\|_A &= \min\left\{\left\|p(M^{-1}A)\, e^{(0)}\right\|_A \,\middle|\, p \in \Pi_k,\ p(0) = 1\right\} \\
&= \min\left\{\left\|\sum_{j=1}^k \gamma^{(j)} p(M^{-1}A) v^{(i^{(j)})}\right\|_A \,\middle|\, p \in \Pi_k,\ p(0) = 1\right\} \\
&= \min\left\{\left\|\sum_{j=1}^k \gamma^{(j)} p(\lambda^{(i^{(j)})}) v^{(i^{(j)})}\right\|_A \,\middle|\, p \in \Pi_k,\ p(0) = 1\right\}
\end{aligned}$$

and we can again use the unique interpolating polynomial of maximum degree k for the $k+1$ points $\{(\lambda^{(i^{(j)})}, 0) \mid j = 1, \dots, k\} \cup \{(0,1)\}$ to obtain a vanishing error after k iterations at the latest. (2 Points)

(*iii*) When

$$M^{-1}r^{(0)} = M^{-1}(Ax^{(0)} - b) = M^{-1}A(x^{(0)} - x^*) = \sum_{j=1}^{k} \gamma^{(j)} v^{(i^{(j)})}$$

for generalized eigenvectors $v^{(i^{(j)})}$ and coefficients $\gamma^{(j)}$ then

$$x^{(0)} - x^* = \sum_{j=1}^{k} \frac{\gamma^{(j)}}{\lambda^{(i^{(j)})}} v^{(i^{(j)})} \in \operatorname{span}\{v^{(i^{(j)})},\ j = 1, \dots, k\}$$

and condition (*ii*) yields the result.

(2 Points)

Homework Problem 3.4 (M-Preconditioned CG for Solving s. p. d. Linear Systems) 6 Points

Implement the M-Preconditioned CG method for solving s. p. d. linear systems as outlined in Algorithm 4.6 of the lecture notes. Additionally include the option of turning conjugacy off (to return to the steepest descent method). Visualize the convergence speed of the method and compare the results to the steepest descent method for pseudo-randomized problems.

Solution.

Since we have already implemented the steepest descent scheme in homework problem 2.4, we can get the implementation of the CG method by very slight modifications (memorizing an additional vector and a scalar plus the orthogonalization step) of the steepest descent implementation.

As we already know, the visualization of the iterates for a two dimensional problem is only partially helpful as the CG method terminates after two iterations at the latest, see Figure 0.2 However, it does give an intuitive understanding of what the A-conjugacy will look like in two dimensions. Generally, increasing an eigenvalue of an s. p. d. matrix will make A conjugate directions be tilted more towards the remaining eigenvalues.

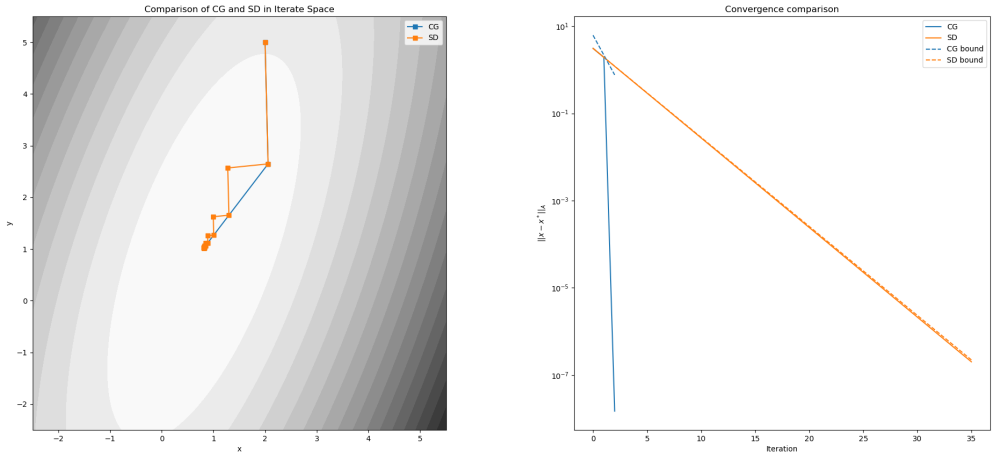

Figure 0.2: Convergence of the steepest descent method (SD) and the conjugate gradient method (CG) in two dimensions.

Now for a randomly generated matrix $A \in \mathbb{R}^{100\times 100}$, with randomly disctributed eigenvalues in $[1, 100]$ with 1 and 100 being eigenvalues (and hence $\kappa = 100$), we can observe much faster convergence of the cg method, as expected, see Figure 0.3. It also converges much fast than the worst case error bound implies and the convergence appears to have a wave-like pattern.

The wave like pattern shows up because the CG method iterates more effectively on eigenvalues that are clustered as indicated by the additional error estimate

$$\|x^{(k+1)} - x^*\|_A \leq \frac{\lambda^{(n-k)} - \lambda^{(1)}}{\lambda^{(n-k)} + \lambda^{(1)}} \|x^{(0)} - x^*\|_A$$

i. e., the strong dips in the error are obtained when a new cluster of eigenvalues is reached. See Figure 0.4.

(6 Points)

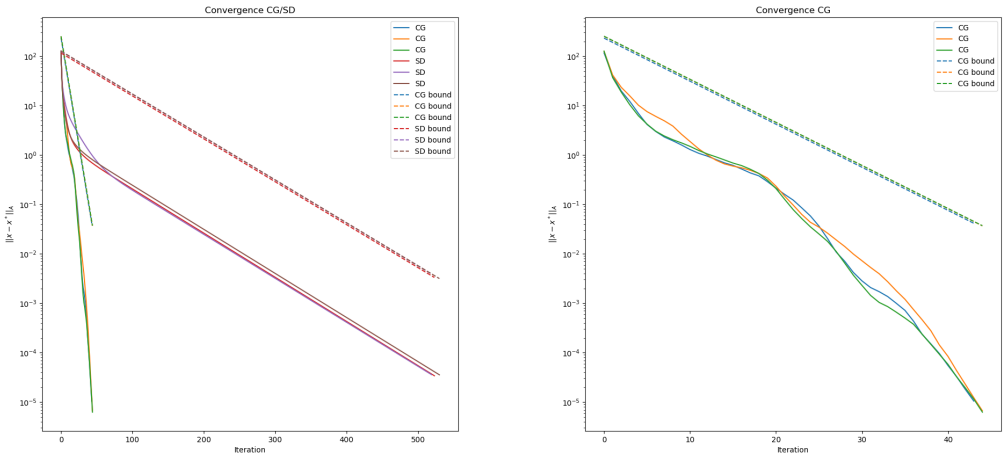

Figure 0.3: Convergence of the conjugate gradient method (CG) and the steepest descent method (SD) for pseudo randomized 100-d problem and three randomized initial guesses. Right hand side shows CG only.

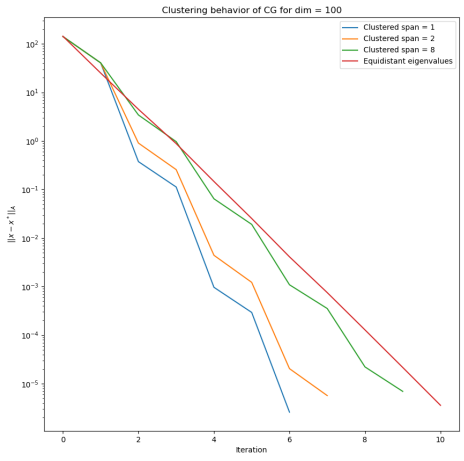

Figure 0.4: Convergence of the conjugate gradient method (CG) for a 100×100 matrix with 2 clusters of eigenvalues and equidistantly distributed eigenvalues in $[1, 100]$ for various upper bound on the maximum distance of eigenvalues in each cluster.

Please submit your solutions as a single pdf and an archive of programs via moodle.