EEE-361 HOMEWORK 2

March 4, 2021

In this homework you will investigate the performance of alternative least squares solvers for $\boldsymbol{Ax} = \boldsymbol{b}$. The numerical techniques to be compared are

- Generalized Minimum Residual (GMRES)
- Conjugate Gradients (CG)
- $\boldsymbol{A}^+$: The Pseudoinverse

The first step is to generate a symmetric matrix $\boldsymbol{A}$ according to the example attached in the following pages (from the book *Numerical Linear Algebra* by Lloyd N. Trefethern and David Bau). You will use the values $\tau = 0.1$ and $\tau = 0.01$ and for each of them you need to create a matrix $\boldsymbol{A}$ with dimensions 100×100, 500×500 and 10000×10000 (if the computations with this size create problems in your computational tools, find the largest matrix size for which you can still obtain the results) as explained in the example. So, in total there will be 6 different matrices $\boldsymbol{A}$.

Now, for each $\boldsymbol{A}$ with dimensions $m \times m$, the next step will be to generate 10 samples of vector $\boldsymbol{x}_0$ by using $\mathsf{randn}(m, 1)$. Note that the vector $\boldsymbol{x}_0 \in \mathbb{R}^m$ is sampled from a standard normal distribution. Then, you need to compute $\boldsymbol{b}_0 = \boldsymbol{Ax}_0$ and subsequently create the vector $\boldsymbol{b} = \boldsymbol{b}_0 + \boldsymbol{w}$, where $\boldsymbol{w} = \sigma_{\boldsymbol{w}} \cdot \mathsf{randn}(m, 1)$ is a vector sampled from a normal distribution with standard deviation $\sigma_{\boldsymbol{w}}$. You will do this for three different values of $\sigma_{\boldsymbol{w}}$, namely $0.0001, 0.01, 1$. So in total, there will be 30 different realizations of the vector $\boldsymbol{b}$, corresponding to the 3 different noise levels. Let us denote them by $\boldsymbol{b}_{i,j}$, for $1 \leq i \leq 3$ and $1 \leq j \leq 10$. Note that the value i is the noise level index and j is the index of the realizations of $\boldsymbol{x}_0$.

At this point, you will employ the three methods in order to approximate the least squares solution $\hat{\boldsymbol{x}}_{i,j}$ to the system $\boldsymbol{Ax} = \boldsymbol{b}_{i,j}$ and define the error vector $\boldsymbol{e}_{i,j} = (-\hat{\boldsymbol{x}}_{i,j} + \boldsymbol{x}_{0,j})$. Now, for $1 \leq i \leq 3$, we define

$$E_{S,i} = \sqrt{\frac{1}{10}\sum_{j=1}^{10} \|\boldsymbol{e}_{i,j}\|_2^2} \quad \text{and} \quad E_{O,i} = \sqrt{\frac{1}{10}\sum_{j=1}^{10} \|\boldsymbol{b}_{i,j} - \boldsymbol{A}\hat{\boldsymbol{x}}_{i,j}\|_2^2}\,.$$

$E_{S,i}$ corresponds to the root-mean squared error on the estimated solution, while $E_{O,i}$ corresponds to the root-mean squared error in the fit to the observations. For each different $\boldsymbol{A}$ and the three methods, you will plot the log values of $E_{S,i}$ with respect to the three values of $\sigma_{\boldsymbol{w}}$. What do you observe? Comment on the results. In the case of the Conjugate Gradients you will also plot the values of $E_{O,i}$ with respect to those of τ, as the plot shown in the example below. Moreover, when you are using the GMRES and Conjugate Gradient approaches:

1. Comment on the stopping criteria you used and investigate what happens if you choose less or significantly more iterations.
2. Check the orthogonality of the bases the algorithms generate for the Krylov subspace along with the iterations.

For efficient solution to the tridiagonal system generated in GMRES algorithm, you can use efficient algorithms such as Tridiagonal matrix algorithm.

Proof. By Theorem 38.3, it is enough to find a polynomial $p \in P_n$ whose maximum value for $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ is the middle expression of (38.10). The polynomial we choose is the scaled and shifted Chebyshev polynomial $p(x) = T_n(\gamma - 2x/(\lambda_{\max} - \lambda_{\min}))/T_n(\gamma)$, where T_n is the usual Chebyshev polynomial of degree n and γ takes the special value $\gamma = (\lambda_{\max} + \lambda_{\min})/(\lambda_{\max} - \lambda_{\min}) = (\kappa + 1)/(\kappa - 1)$. For $x \in [\lambda_{\min}, \lambda_{\max}]$, the argument of T_n in the numerator of $p(x)$ lies in $[-1, 1]$, which means the magnitude of that numerator is ≤ 1. Therefore, to prove the theorem, it will suffice to show

$$T_n(\gamma) = T_n\left(\frac{\kappa+1}{\kappa-1}\right) = \frac{1}{2}\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^n + \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^{-n}\right]. \qquad (38.11)$$

We can do this by making the change of variables $x = \frac{1}{2}(z + z^{-1})$, $T_n(x) = \frac{1}{2}(z^n + z^{-n})$, standard in the study of Chebyshev polynomials. If $(\kappa+1)/(\kappa-1) = \frac{1}{2}(z + z^{-1})$, that is, $\frac{1}{2}z^2 - (\kappa+1)/(\kappa-1)z + \frac{1}{2} = 0$, then we have a quadratic equation with solution

$$z = \left(\frac{\kappa+1}{\kappa-1}\right) + \sqrt{\left(\frac{\kappa+1}{\kappa-1}\right)^2 - 1} = \frac{\kappa+1+\sqrt{(\kappa+1)^2-(\kappa-1)^2}}{\kappa-1}$$

$$= \frac{\kappa+1+\sqrt{4\kappa}}{\kappa-1} = \frac{(\sqrt{\kappa}+1)^2}{(\sqrt{\kappa}+1)(\sqrt{\kappa}-1)} = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}.$$

Thus $T_n(\gamma) = \frac{1}{2}(z^n + z^{-n})$ for this value of z, which is (38.11), as claimed. □

Theorem 38.5 is the most famous result about convergence of the CG iteration. Since

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \sim 1 - \frac{2}{\sqrt{\kappa}}$$

as $\kappa \to \infty$, it implies that if κ is large but not too large, convergence to a specified tolerance can be expected in $O(\sqrt{\kappa})$ iterations. One must remember that this is only an upper bound. Convergence may be faster for special right-hand sides (not so common) or if the spectrum is clustered (more common).

Example

For an example of the convergence of CG, consider a 500×500 sparse matrix A constructed as follows. First we put 1 at each diagonal position and a random number from the uniform distribution on $[-1, 1]$ at each off-diagonal position (maintaining the symmetry $A = A^T$). Then we replace each off-diagonal entry with $|a_{ij}| > \tau$ by zero, where τ is a parameter. For τ close to zero, the result is a well-conditioned positive definite matrix whose density of nonzero entries is approximately τ. As τ increases, both the condition number and the sparsity deteriorate.

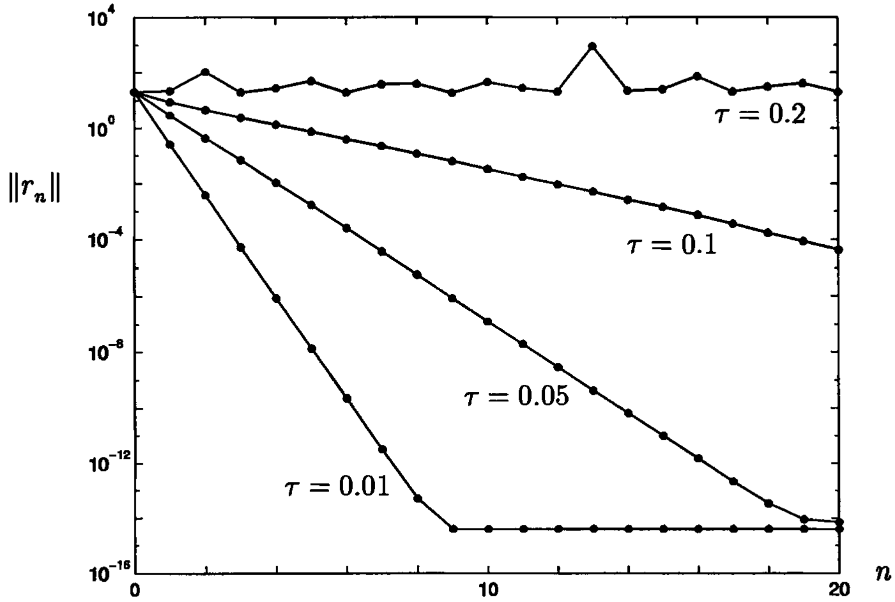

Figure 38.1. *CG convergence curves for the* 500 × 500 *sparse matrices* A *described in the text. For* $\tau = 0.01$, *the system is solved about* 700 *times faster by CG than by Cholesky factorization. For* $\tau = 0.2$, *the matrix is not positive definite and there is no convergence.*

Figure 38.1 shows convergence curves corresponding to 20 steps of the CG iteration for matrices of this kind with $\tau = 0.01, 0.05, 0.1, 0.2$. (The right-hand side b was taken to be a random vector.) For $\tau = 0.01$, A has 3092 nonzero entries and condition number $\kappa \approx 1.06$. Convergence to machine precision takes place in 9 steps, about 6×10^4 flops. For $\tau = 0.05$, there are 13,062 nonzeros with $\kappa \approx 1.83$, and convergence takes 19 steps, about 5×10^5 flops. For $\tau = 0.1$ we have 25,526 nonzeros and $\kappa \approx 10.3$, with only 5 digits of convergence after 20 steps and 10^6 flops. For $\tau = 0.2$, with 50,834 nonzeros, there is no convergence at all. The lowest eigenvalue is now negative, so A is no longer positive definite and the use of the CG iteration is inappropriate. (In fact, the CG iteration often succeeds with indefinite matrices, but in this case the matrix is not only indefinite but ill-conditioned.)

Note how closely the $\tau = 0.01$ curve of Figure 38.1 matches the schematic ideal depicted in Figure 32.1! For this example, the operation count of 6×10^4 flops beats Cholesky factorization (23.4) by a factor of about 700. Unfortunately, not every matrix arising in practice has such a well-behaved spectrum, even after the best efforts to find a good preconditioner.