CS 6210: Homework 3

Instructor: Anil Damle
Due: October 9, 2024

Policies

You may discuss the homework problems freely with other students, but please refrain from looking at their code or writeups (or sharing your own). Ultimately, you must implement your own code and write up your own solution to be turned in. Your solution, including plots and requested output from your code must be typeset and submitted via the Gradescope as a pdf file. Additionally, please submit any code written for the assignment. This can be done by either including it in your solution as an appendix, or uploading it as a zip file to the separate Gradescope assignment.

Question 1:

Let us consider a symmetric, but possibly indefinite, matrix $A \in \mathbb{R}^{n\times n}$. While we cannot appeal to a Cholesky decomposition when A is indefinite, it may seem natural to ask if we can still get a "symmetric" factorization. It turns out that we can. In particular, we can seek a factorization of the form $A = LDL^T$ where L is unit lower triangular and D is diagonal. We can motivate the existence of such a factorization via the observation that if $a_{11} \neq 0$ then we have the identity

$$A = \begin{bmatrix} a_{11} & v^T \\ v & A_2 \end{bmatrix} = \begin{bmatrix} 1 & \\ \frac{v}{a_{11}} & I \end{bmatrix} \begin{bmatrix} a_{11} & \\ & A_2 - \frac{vv^T}{a_{11}} \end{bmatrix} \begin{bmatrix} 1 & \frac{v^T}{a_{11}} \\ & I \end{bmatrix},$$

where $v = A(2:n,1)$ and $A_2 = A(2:n,2:n)$. We could then recurse on $A_2 - \frac{vv^T}{a_{11}}$ to get an LDL^T factorization of A. More concretely, given $A_2 - \frac{vv^T}{a_{11}} = LDL^T$ we have that

$$A = \begin{bmatrix} 1 & \\ \frac{v}{a_{11}} & L \end{bmatrix} \begin{bmatrix} a_{11} & \\ & D \end{bmatrix} \begin{bmatrix} 1 & \\ \frac{v}{a_{11}} & L \end{bmatrix}^T.$$

However, as with LU factorization, directly following this path may be treacherous, e.g., if $|a_{11}|$ is small. So, we might want to introduce pivoting. However, to retain symmetry we must use symmetric pivoting and that only allows us to consider pivots on the diagonal. Specifically, at the first step we would choose a permutation P_1 such that the largest magnitude entry on the diagonal of $P_1 A P_1^T$ is in the $(1,1)$ location. We would then compute P_2 to move the largest magnitude diagonal entry of $A_2 - \frac{vv^T}{a_{11}}$ to the $(1,1)$ location and so on (analogously to LU).

Unfortunately, this scheme is not sufficient to stabilize LDL^T factorization. In general we must allow D to be block diagonal with 1×1 and 2×2 blocks to accomplish that goal. A quick way to see this is that

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

has no LDL^T factorization with diagonal D. Similar to LU, the issues persist even if the diagonal is not exactly zero but small. The goal of this problem is (mostly) to exercise your understanding of the structure of triangular factorizations, so we will ignore this issue.[1]

[1] There are classes of matrices where it is fine to only consider diagonal D, so this is not completely unreasonable.

Write code that computes an LDL^T factorization with symmetric pivoting given a symmetric $A \in \mathbb{R}^{n\times n}$. Then use your code to address the following:

(a) Provide pseudo code for your implementation.

(b) What is the asymptotic computational complexity of this factorization? Demonstrate that your code achieves the expected complexity.[2]

(c) Consider computing the LDL^T factorization of the so-called Hilbert matrix of size n (defined as $H_{ij} = 1/(i+j-1)$ for $i, j = 1, \ldots, n$). Report L, D, and P to at least 4 significant digits for the 4×4 Hilbert matrix.

(d) For the Hilbert matrix, what do you observe about $\|LDL^T - PHP^T\|_2/\|H\|_2$ for moderate n? What about the accuracy of a solution to $Hx = b$ as quantified by the relative residual? Explain your observations.

(e) Since the Hilbert matrix H is symmetric positive definite (this is a fun proof to try if you are so inclined), presumably it would be natural to use a Cholesky decomposition rather than an LU factorization. Repeat the previous part using a built-in Cholesky decomposition routine (or implement it yourself if you like). What do you observe?

QUESTION 2:

Assume that we are using a stationary iterative method to solve $Ax = b$ with the splitting $A = M - N$ and initial guess $x^{(0)}$, and that $x^{(1)}, \ldots, x^{(k)}$ have been computed using the iteration $Mx^{(j+1)} = Nx^{(j)} + b$. Normally we would consider $x^{(k)}$ as our current approximation of the solution $x = A^{-1}b$. However, maybe there is some process that allows us to accelerate the convergence of our method and draw an interesting connection to Krylov subspace methods.

Let's consider one approach to accomplishing this goal. Specifically, we would like to construct coefficients $\{v_j^{(k)}\}_{j=1}^k$ for each iteration k such that

$$y^{(k)} = \sum_{j=0}^{k} v_j^{(k)} x^{(j)}$$

is, hopefully, a better approximation of our true solution x than $x^{(k)}$. Define $G = M^{-1}N$ and

$$p_k(z) = \sum_{j=0}^{k} v_j^{(k)} z^j.$$

(a) Assuming $x^{(0)} = 0$, is $y^{(k)}$ consistently part of a Krylov subspace? If so, which Krylov subspace? (By consistently, I mean can you define a matrix H and vector w that do not depend on k such that $y^{(k)} \in \mathcal{K}_k(H, w)$.)

(b) Now, let us further assume that $p_k(1) = 1$. Prove that

$$y^{(k)} - x = p_k(G)\, e^{(0)}$$

[2]If you need test matrices that will avoid the aforementioned issues you can choose A to be strictly diagonally dominant with positive diagonal entries, i.e., $a_{ii} > \sum_j \neq i |a_{ij}|$.

(c) Prove that if B is similar to a Hermitian matrix then

$$\rho(p_k(B)) = \max_{\lambda_i \in \lambda(B)} |p_k(\lambda_i)|$$

where $\rho(p_k(B))$ is the spectral radius of $p_k(B)$.

We now assume that the iteration matrix G is similar to a Hermitian matrix and has real eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$. Recall that for convergence (from any initial guess) we require that

$$-1 < \lambda_n \leq \lambda_1 < 1.$$

Moreover, let α and β be such that

$$-1 < \alpha \leq \lambda_n \leq \cdots \leq \lambda_1 \leq \beta < 1.$$

Since we may write

$$\max_{\lambda_i \in \lambda(B)} |p_k(\lambda_i)| \leq \max_{\alpha \leq \lambda \leq \beta} |p_k(\lambda)|,$$

it seems reasonable to pick p_k to be small on the interval $[\alpha, \beta]$. The ideal choice, given the constraint on $p_k(1)$, is a scaled and shifted version of the k^{th} Chebyshev polynomial. These polynomials may be defined by the recursion $c_j(z) = 2zc_{j-1}(z) - c_{j-2}(z)$ where $c_0(z) = 1$ and $c_1(z) = z$. Alternatively, we may write $c_j(z) = \cos(j\theta)$ where $\theta = \arccos(z)$. Specifically, we may choose the polynomial

$$p_k(z) = \frac{c_k\left(-1 + 2\frac{z-\alpha}{\beta-\alpha}\right)}{c_k(\mu)},$$

where $\mu = 1 + 2\frac{1-\beta}{\beta-\alpha}$. You can verify that $p_k(1) = 1$. Notably, $c_k(z)$ has the property that it is bounded between -1 and 1 in the interval $[-1, 1]$, but then grows rapidly outside of this interval. So, in the formula above, $c_k(\mu)$ becomes large as $k \to \infty$. With the chosen scaling we ensure that $p(z)$ is small in the interval $[\alpha, \beta]$ while satisfying $p(1) = 1$.

(d) Given the above choice for $p_k(z)$ prove that there exists a constant C such that

$$\|y^{(k)} - x\|_2 \leq C\left(\frac{1}{c_k(\mu)}\right)\|x - x^{(0)}\|_2,$$

where C may depend on the matrix G.

(e) Let $\alpha = -0.9$ and $\beta = 0.9$. Plot $c_k(\mu)$, on a logarithmic scale, for $k = 0, 1, \ldots, 100$.

We will now consider using this acceleration method in conjunction with the Jacobi iteration. For the remainder of this problem assume that A is a real symmetric matrix that is strictly diagonally dominant and has positive diagonal entries.

(f) Under the aforementioned assumptions, prove that the iteration matrix associated with A is similar to a Hermitian matrix.

(g) Implement the Jacobi method both with and without Chebyshev acceleration. You can find pseudo-code in Golub and Van Loan 4th edition, section 11.2.8 (3rd edition, section 10.1.5) that leverages the three-term recurrence for Chebyshev polynomials for an efficient implementation.

Build some non-singular test problems via matrices A, and vectors $x^{(0)}$ and b where the eigenvalues of G approach ± 1. You may use a built in routine to compute eigenvalues and set the eigenvalue bounds as $\alpha = (-1 + \lambda_n)/2$ and $\beta = (1 + \lambda_1)/2$. Use your algorithm both with and without the acceleration to solve $Ax = b$. You may stop your algorithm when the 2 norm of the relative residual is less than 10^{-6} or you have run 1000 iterations. Provide error plots, on a logarithmic scale, of the 2 norm of the residual vs iteration both with and without the acceleration. Comment on your observations.

QUESTION 3 (UNGRADED, BUT INTERESTING):

In class we saw a backwards stability style result for pivoted LU factorization that incorporated the growth factor ρ. You may have noticed that we did not see any forward error results, i.e., we never said anything about $\tilde{L} - L$ or $\tilde{U} - U$. In fact, the forward error can be quite large. However, often what we ultimately care about is the solution to $Ax = b$. Devise a backwards error bound for solving $Ax = b$ with non-singular A using LU with partial pivoting followed by a sequence of triangular solves.[3] Your bound should explicitly incorporate the growth factor ρ. For this problem you may assume that in exact arithmetic there are no ties in the pivoting procedure and μ is sufficiently small such that the computed permutation matches the exact one.

One reason this problem is ungraded is that it has a very subtle point burried within it. While the goal of this question is for you to see how backwards error results can be "chained" together to get a result for the solution to the linear system, it requires bounds on $\|\tilde{L}\|$ and $\|\tilde{U}\|/\|A\|$.

In class we saw that when using partial pivoting we can ensure $\|L\| = \mathcal{O}(1)$ because all of the entries have magnitude bounded by 1. For $\tilde{L}$ we can make a similar argument, though formally the easiest path is to ensure all the entires have magnitude less than $1 + \mu$ (which does not really change anything).

Unfortunately, bounding $\|\tilde{U}\|/\|A\|$ is more complex. In particular we (like many, but not all, books) defined the growth factor in terms of the exact U. Therefore, it is slightly delicate to use it directly for bounding $\|\tilde{U}\|/\|A\|$. For the purposes of this problem it suffices to derive a bound in terms of both the "exact" growth factor ρ and a computed growth factor $\tilde{\rho}$ (defined as the growth factor realized by $\tilde{U}$).

This is actually a rather annoying sticking point in the analysis of these algorithms, and I am certainly happy to discuss it further. For example, to quote Higham (Accuracy and Stability of Numerical Algorithms; Section 9.3, page 165) when providing a theorem on backward error for solving $Ax = b$ via partially pivoted LU (Note, that in this book $\widehat{L}$ and $\widehat{U}$ are the computed LU factors):

> "We hasten to admit to using an illicit manoeuvre in the derivation of this theorem: we have used bounds for $\widehat{L}$ and $\widehat{U}$ that strictly are valid only for the exact L and U."

Higham goes on to comment (Accuracy and Stability of Numerical Algorithms; Section 9.14, page 189):

> "The dilemma of whether to define the growth factor in terms of exact or computed quantities is faced by all authors; most make one choice or the other, and go on to derive, without comment, bounds that are strictly incorrect."

So, there you have that.

[3] The computed solution $\tilde{x}$ solves a linear system $(A + \delta A)\tilde{x} = b$, what can you say about $\|\delta A\|/\|A\|$?