MA2501 Numerical Methods
Spring 2017

Solutions to exercise set 4

Norwegian University of Science
and Technology
Department of Mathematics

$\boxed{1}$ **a)** Compute the LDL^T and, if possible, Cholesky factorization of the following matrices:

$$\mathbf{A}_1 = \begin{bmatrix} -2 & -4 & 6 & -2 \\ -4 & -5 & 12 & -7 \\ 6 & 12 & -17 & 7 \\ -2 & -7 & 7 & 1 \end{bmatrix}, \qquad \mathbf{A}_2 = \begin{bmatrix} 1 & 2 & -3 & 1 \\ 2 & 7 & -6 & -1 \\ -3 & -6 & 10 & -2 \\ 1 & -1 & -2 & 6 \end{bmatrix}.$$

b) Use the previously computed factorizations to solve the linear systems $\mathbf{A}_j x = \mathbf{b}$ with $\mathbf{b} = [-3, 3, 1, -3]^T$.

Possible solution:

a) The LDL^T factorization of the matrix $\mathbf{A}_1$ uses the matrices

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 1 & -1 & 1 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{D} = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix};$$

it does not have a Cholesky factorization, as it is not positive definite (the entry a_{11} is negative).

For the matrix $\mathbf{A}_2$ we have an LDL^T factorization with matrices

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 1 & -1 & 1 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Moreover, it has a Cholesky factorization with the matrix

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & \sqrt{3} & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 1 & -\sqrt{3} & 1 & 1 \end{bmatrix}.$$

b) The first system has the solution $\mathbf{x} = [73.5, -14, 9, -17]^T$, the second system the solution $\mathbf{x} = [-135, 20, -25, 17]^T$.

In the case of the LDL^T factorization, the first solution is obtained by first solving the equation $\mathbf{Ly} = \mathbf{b}$, which yields the result $\mathbf{y} = [-3, 9, -8, 17]^T$ (note that the system is unit triangular, thus one can in a sense read off the result using the formula

$y_k = b_k - \sum_{j=1}^{k-1} \ell_{kj} y_j$). Then one solves the system $\mathbf{Dz} = \mathbf{y}$ (which means: one computes $z_i = y_i/d_{ii}$), which yields the result $\mathbf{z} = [1.5, 3, -8, -17]^T$. Finally, one computes $\mathbf{x}$, starting from x_n, with the formula $x_k = z_k - \sum_{j=k+1}^{n} \ell_{jk} x_j$.

In the second case, one obtains the intermediate results $\mathbf{y} = [-3, 9, -8, 17]^T$ and $\mathbf{z} = [-3, 3, -8, 17]^T$.

For the Cholesky factorization, the approach is similar. Here one solves first the equation $\mathbf{Ly} = \mathbf{b}$ using the formula $y_k = (b_k - \sum_{j=1}^{k-1} \ell_{kj} y_j)/d_{kk}$ and then $\mathbf{L}^T\mathbf{x} = \mathbf{y}$ using $x_k = (y_k - \sum_{j=k+1}^{n} \ell_{jk} x_j)/d_{jj}$. Here one obtains the intermediate result $\mathbf{y} = [-3, \sqrt{27}, -8, 17]^T$.

$\boxed{2}$ How many operations are required for the computation of the Cholesky factorization of a symmetric and positive definite matrix? Compare the result with the numerical complexity of the LU factorization.

Possible solution:

The diagonal entries of $\mathbf{L}$ are given by the formula

$$\ell_{ii} = \Big(a_{ii} - \sum_{j=1}^{i-1} \ell_{ij}^2\Big)^{1/2}.$$

Thus the i-th diagonal entry requires $i-1$ additions, $i-1$ multiplications, and taking one square root. For the off-diagonal entries we have the formula

$$\ell_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} \ell_{ik}\ell_{jk}}{\ell_{jj}}.$$

In column j, each entry requires therefore $j-1$ additions, $j-1$ multiplications, and one division. Since column j has $n-j+1$ entries, it thus needs $(j-1)(n-j+1)$ additions and multiplications. Summing over all columns, we thus obtain

$$\sum_{j=1}^{n} (j-1)(n-j+1) = \sum_{j=1}^{n} n(j-1) - (j-1)^2 = n\frac{n(n-1)}{2} - \frac{n(n-1)(2n-1)}{6} \approx \frac{n^3}{6}$$

multiplications and the same number of additions. Moreover we have $n(n-1)/2$ divisions (one for each off-diagonal entry), and n square root computations (one for each diagonal entry). We can assume that taking square roots is not much more complicated than multiplication[1] Consequently, the total number of long operations is approximately $n^3/6$, which is about half the number that is needed for the computation of an LU factorization.

$\boxed{3}$ Write a Matlab program for the solution of a linear system $\mathbf{Ax} = \mathbf{b}$ with symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ using the LDL^T factorization. Include some error messages for the situation that $\mathbf{A}$ is non-symmetric or the factorization fails.

[1] One can, for instance, use the *Babylonian method* (which is nothing else than Newton's method for solving the equation $x^2 - a = 0$). This method basically needs a fixed number of additions and divisions for obtaining a result of a given precision. More details can be found here.

Possible solution:

Example code can be found on the webpage of the course.

4 Use the bisection method for finding a solution of the equation $x^2 - 2 = 0$ (i.e., for computing $\sqrt{2}$) starting with the interval $[1, 2]$. Stop the computations when the error is smaller than 10^{-3}. How many iterations are needed to obtain results with an error smaller than 10^{-12}?

Possible solution:

We obtain the intervals

$$\begin{aligned}
a &= 1, & b &= 2,\\
a &= 1, & b &= 1.5,\\
a &= 1.25, & b &= 1.5,\\
a &= 1.375, & b &= 1.5,\\
a &= 1.375, & b &= 1.4375,\\
a &= 1.40625, & b &= 1.4375,\\
a &= 1.40625, & b &= 1.421875,\\
a &= 1.4140625, & b &= 1.421875,\\
a &= 1.4140625, & b &= 1.41796875,\\
a &= 1.4140625, & b &= 1.416015625,\\
a &= 1.4140625, & b &= 1.4150390625.
\end{aligned}$$

After that, the length of the solution interval is smaller than 10^{-3}.

In general, the error after the k-th step is smaller than 2^{-k} times the length of the first interval (which is 1). Thus we will need more than $\log_2(10^{12}) \approx 39.9$ steps for an accuracy of 10^{-12}. In other words: 40 iterations are required for the desired accuracy.

5 Implement the bisection method in Matlab. Your program should take as an input the two initial points $a < b$, the desired accuracy $\varepsilon > 0$, and a reference to the function f a root of which is to be computed.

Use your program for finding the solution of the equation $\tan(x) = 0$ in the interval $[2, 4]$ (which should be π) and a solution of the equation $3\cos(x)\sin(x^2) = 1$ in the interval $[0, 10]$. What happens if you apply the method to the equation $\tan(x) = 0$ with the starting interval $[1, 2]$?

Possible solution:

You find a possible implementation on the webpage of the course.

Note that the method will “fail” in case of the function tan on the interval $[1, 2]$. More precisely, the search interval will shrink to the point $\pi/2$, where tan has a singularity (and report the limit as a solution of the equation). One possibility to safeguard against this behavior is to include error (or warning) messages in case the function values appear not to converge to 0.

$\boxed{6}$ Suppose $f(x)$ is $C[a, b]$ with a sign change on and unique zero in the interior of $[a, b]$. Let $h(x) = f(x)g(x)$ where $g(x)$ is a positive and continous function on $[a, b]$. Let c_f^n and c_h^n be the n-th approximation from the Bisection method to the root of f and h respectively, with initial interval $[a, b]$. Can we say anything about c_f^n compared to c_h^n?

Possible solution:

The Bisection method assumes a continous function that changes sign on an interval $[a, b]$. If such a function has a unique zero on this interval, the Bisection method finds an approximation to it. The method repeatedly halves the initial interval, always choosing the half where the function changes sign. It continues until it stumbles upon the actual root or until the current interval is small enough. The approximation after n iterations c_n is the midpoint of the current interval $[a_n, b_n]$. From the description it is clear that the series of approximations $\{c_n\}_{n=0}^{\infty}$ generated by the method only depends upon the sign and zero of the function on $[a, b]$. Since the product of continous functions is again continous and $g(x)$ is positive on $[a, b]$, it follows that $f(x)$ and $h(x)$ are both continous and have the same sign function (same sign and same zero) on $[a, b]$. Thus both functions will satisfy the assumptions for the Bisection method to be applicable, and $c_n^f = c_n^h$ for all $n \geq 0$