POLITECNICO
MILANO 1863

Course of Numerical Methods for Engineering
Lab 6

Luca Bonaventura, Tommaso Benacchio

MOX - Politecnico di Milano

PHYS-ENG, A.Y. 2020-21

19-20/10/2020

Topics of this session:

- **Cholesky factorization**
- **Sparse matrix representation and matrix factorizations**

Cholesky factorization

Definition

A matrix A is called positive definite if for any nonzero vector x one has $x^T Ax > 0$.

Theorem

A symmetric matrix is positive definite if it has only positive eigenvalues.

Theorem

Given an $n \times n$ symmetric and positive definite matrix A, there is a unique lower triangular matrix L such that $l_{i,i} > 0,\ i = 1, \ldots, n$ for which the Cholesky factorization $A = LL^T$ holds. The matrix L is also called Cholesky factor of A.

Exercise 1

Given the matrix

$$A = \begin{pmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 5 & 4 & 3 & 2 \\ 3 & 4 & 5 & 4 & 3 \\ 2 & 3 & 4 & 5 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

consider the linear system $Ax = b$ with a right hand side such that its exact solution is given by the vector $x_{ex} = [1, 2, 3, 4, 5]^T$. Check that A is symmetric and positive definite (use the command `eig`). Solve the system

(a) **with the LU factorization, determining whether or not pivoting has been carried out and using \ (backslash) command to solve the resulting lower triangular and upper triangular systems**

(b) **with the Cholesky factorization, using the command `chol` to determine the factorization and the \ (backslash) command to solve the resulting lower triangular and upper triangular systems; check that the Cholesky L factor is different from that provided by the LU factorization;**

Compute in both cases the absolute and relative error of the computed solution with respect to x_{ex} in the l^2 and l^∞ norm.

Exercise 2

a) **Repeat the steps in exercise 1 for the matrix** $A = H_{13}$, **where** H_n **denotes the** $n \times n$ **Hilbert matrix built with the command** `hilb`, **and considering consider the linear system** $Ax = b$ **with a right hand side such that its exact solution is given by the vector** $x_{ex} = [-1, -1, \ldots, -1, -1]^T$.

b) **repeat the above for** $n = 14$;

c) **repeat the above for** $A = I + H_{14}$.

Exercise 3

Consider a $n \times n$ matrix A with values equal to 5 on the main diagonal and to to -1 on the first and second superdiagonals and subdiagonals. Assuming $n = 50, 500, 5000$, build the matrix both in full and in sparse format and consider the linear system $Ax = b$ with a right hand side such that its exact solution is given by the vector $x_{ex} = [1, 1, \ldots, 1, 1]^T$.
Solve the linear systems associated to the matrix in the two formats:

(a) **with the LU factorization, determining whether or not pivoting has been carried out and using the \ (backslash) command to solve the resulting lower triangular and upper triangular systems**

(b) **with the Cholesky factorization, using the command `chol` to determine the factorization and the \ (backslash) command to solve the resulting lower triangular and upper triangular systems ; check that the Cholesky L factor is different from that provided by the LU factorization;**

Compute the absolute and relative error of the computed solution with respect to x_{ex} in the l^2 and l^∞ norm; compare the computational effort required by the two methods in each case, estimated using the `tic` and `toc` commands.

Cholesky decomposition algorithm

Definition

Given an $n \times n$ symmetric and positive definite matrix A, the unique lower triangular matrix L defining the Cholesky decomposition can be computed as follows:

- $l_{1,1} = \sqrt{a_{1,1}}$;
- **for $i = 2, \ldots, n$ set**

$$
\begin{aligned}
l_{i,j} &= \frac{a_{i,j} - \sum_{k=1}^{j-1} l_{i,k} l_{j,k}}{l_{j,j}} \qquad j = 1, \ldots, i-1; \\
l_{i,i} &= \sqrt{a_{i,i} - \sum_{k=1}^{i-1} l_{i,k}^2} \qquad j = i.
\end{aligned}
$$

Exercise 4

Write a MATLAB function implementing the Cholesky decomposition algorithm for full matrices. Write a specific variant for banded matrices, in which only the non zero terms of the Cholesky factor are computed.

Banded matrices and factorization

Definition

A matrix A whose entries $a_{i,j}$ are such that $a_{i,j} = 0$ if $j > i + p$ or if $j < i - p$ is called banded. The integer p is called the matrix bandwidth. Banded matrices with bandwidth 1 are called tridiagonal.

The LU factors of the tridiagonal matrix

$$\mathrm{A} = \begin{bmatrix} d_1 & f_1 & 0 & \dots & 0 \\ e_2 & d_2 & f_2 & \dots & 0 \\ 0 & \dots & \dots & \dots & \\ \dots & \dots & 0 & d_{n-1} & f_{n-1} \\ 0 & \dots & 0 & e_{n-1} & d_n \end{bmatrix},$$

can be computed with the following procedure:

- **set** $u_1 = d_1$;
- **for** $i = 2, \dots, n$ **set**

$$l_i = e_i / u_{i-1} \qquad u_i = d_i - f_{i-1} l_i.$$

Sparse matrices

Definition

A $n \times n$ matrix A for which the number of non zero entries $a_{i,j}$ is $O(n)$ is called sparse.

In Octave/MATLAB, two storage classes and functions: `full` and `sparse`

Storage requirement for an $m \times n$ real sparse matrix with nnz nonzero entries: is nnz reals and $nnz + n$ integers.

Sparsity rules in Octave/MATLAB

- **Functions that accept a matrix and return a scalar, constant-size vector, or matrices, such as `size`, `zeros`, `ones`, `rand`, and `eye`, always produce output in full storage format.**
- **The sparse analog of `zeros(m,n)` is simply `sparse(m,n)`. The sparse analogs of `rand` and `eye` are `sprand` and `speye`. There is no sparse analog for the other functions at the previous point.**
- **Unary functions that accept a matrix and return matrices or vectors preserve the storage class of the operand: if `S` is a sparse matrix, then `lu(S)`, `chol(S)` return sparse matrices**
- **Binary operators yield sparse results if both operands are sparse, and full results if both are full. For mixed operands, the result is full unless the operation preserves sparsity. If `S` is sparse and `F` is full, then `S+F`, `S*F`, and `F\S` are full, while `S.*F` is sparse**

Exercise 5

Write a MATLAB function that solves a diagonally dominant tridiagonal linear system with the specific LU factorization algorithm for tridiagonal systems and by adapting the forward elimination and backward substitution to this special case.

(a) **Apply the function to solve a $n \times n$ linear system with associated matrix A with values equal to 4 on the main diagonal, to 1 on the first superdiagonal and -1 on the first subdiagonal and with a right hand side such that its exact solution is given by the vector $x_{ex} = [2, 2, \ldots, 2, 2]^T$.**

(b) **In the case $n = 100$, compare the computational effort required by the method with that of the solution based on standard LU factorization, estimated using the `tic` and `toc` commands. Repeat the exercise for $n = 1000, 10000$.**

Exercise 6

Repeat Exercise 5 using the matrix A in sparse storage form. Use the MATLAB command `spdiags` to build matrix A in sparse storage form. Use the standard LU factorization method and the backslash command $\backslash$ to solve the triangular systems. Compare the execution times with those obtained in Exercise 1, and additionally consider the cases $n = 10^5, 10^6, 10^7$.

Exercise 7

Build the $n \times n$ matrix A that has elements equal to 10 on the main diagonal, 1 on the first and last rows and columns (outside the main diagonal), and -1 on the superdiagonal $n/2$ (leaving the first and last element of that superdiagonal equal to 1). Represent the matrix in sparse storage format.

(a) **Consider $n = 100$. Define $\mathrm{b} = [1, -1, \ldots, 1, -1]^T \in \mathbb{R}^{100}$ and solve the system $\mathrm{Ax} = \mathrm{b}$ using the LU factorization method, both in the sparse storage form and in the full storage form. Check that the results are the same and compare the execution times required for the solution by the two different storage formats.**

(b) **Repeat the exercise for the analogous $n = 500$ and 2000 matrix.**

(c) **For all the above matrices, compare the execution times of the solution with the backslash $\backslash$ command.**