STAT 309: MATHEMATICAL COMPUTATIONS I
FALL 2023
LECTURE 13

1. PIVOTING STRATEGIES

- the (k, k) entry at step k during Gaussian elimination is called the *pivoting entry* or just *pivot* for short
- in the preceding section, we said that if the pivoting entry is zero, i.e., $a_{kk}^{(k)} = 0$, then we just need to find an entry below it in the same column, i.e., $a_{ik}^{(k)}$ for some $i > k$, and then permute this entry into the pivoting position, before carrying on with the algorithm
- but it is really better to choose the *largest* entry below the pivot, and not just any nonzero entry
- that is, the permutation Π_k is chosen so that row k is interchanged with row i, where $|a_{ik}^{(k)}| = \max_{i=k,k+1,\dots,n} |a_{ik}^{(k)}|$, i.e., upon this permutation, we are guaranteed
$$|a_{kk}^{(k)}| = \max_{i=k,k+1,\dots,n} |a_{ik}^{(k)}|$$
- this guarantees that $|\ell_{kj}| \le 1$ for all k and j
- this strategy is known as *partial pivoting*, which is guaranteed to produce an LU factorization if $A \in \mathbb{R}^{m\times n}$ has full column-rank, i.e., $\operatorname{rank}(A) = n \le m$ (it can fail if A doesn't have full column-rank, think of what happens when A has a column of zeros)
- another common strategy, *complete pivoting*, which uses both row and column interchanges to ensure that at step k of the algorithm, the element $a_{kk}^{(k)}$ is the largest element in absolute value from the entire submatrix obtained by deleting the first $k-1$ rows and columns, i.e.,
$$|a_{kk}^{(k)}| = \max_{\substack{i=k,k+1,\dots,n\\ j=k,k+1,\dots,n}} |a_{ij}^{(k)}|$$
- in this case we need both row and column permutation matrices, i.e., we get
$$\Pi_1 A \Pi_2 = LU$$
when we do complete pivoting
- complete pivoting is necessary when $\operatorname{rank}(A) < \min\{m, n\}$
- the factor
$$\gamma_n := \frac{\max_{i,j,k} a_{ij}^{(k)}}{\max_{i,j} a_{ij}}$$
is called the *growth factor* and it quantifies how much the size of the entries grow through the algorithm
- for partial pivoting,
$$\gamma_n^{\text{GEPP}} = 2^{n-1}$$

Date: November 13, 2023, version 1.0. Comments, bug reports: `lekheng@uchicago.edu`.

note that this is a worst case bound, attained by an $n \times n$ matrix of the form (shown below for $n = 5$)

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

- nevertheless in practice the growth in GEPP is pretty small, which is why it is still one of the most widely used algorithm in all of science and engineering
- Wilkinson gave a bound for the *growth factor for complete pivoting*

$$\gamma_n^{\text{GECP}} \le (2 \cdot 3^{1/2} \cdot \dots \cdot n^{1/(n-1)} \cdot n)^{1/2}$$

  the right-hand side is roughly $cn^{\frac{1}{2}} n^{\frac{1}{4}\log n}$ but it is known that this is not the best possible bound
- until 1990, it was conjectured that $\gamma_n^{\text{GECP}} \le n$
- it was shown to be true for $n \le 5$, but there have been examples constructed for $n > 5$ where $\gamma_n^{\text{GECP}} > n$
- there are yet other pivoting strategies due to considerations such as preserving sparsity (if you're interested, look up *minimum degree algorithm* or *Markowitz algorithm*) or a tradeoff between partial and complete pivoting (e.g., *rook pivoting*)

2. UNIQUENESS OF THE LU FACTORIZATION

- the LU decomposition of a nonsingular matrix, if it exists (i.e., without row or column permutations), is unique
- if A has two LU decompositions, $A = L_1U_1$ and $A = L_2U_2$
- from $L_1U_1 = L_2U_2$ we obtain $L_2^{-1}L_1 = U_2U_1^{-1}$
- the inverse of a unit lower triangular matrix is a unit lower triangular matrix, and the product of two unit lower triangular matrices is a unit lower triangular matrix, so $L_2^{-1}L_1$ must be a unit lower triangular matrix
- similarly, $U_2U_1^{-1}$ is an upper triangular matrix
- the only matrix that is both upper triangular and unit lower triangular is the identity matrix I, so we must have $L_1 = L_2$ and $U_1 = U_2$

3. GAUSS–JORDAN ELIMINATION

- a variant of Gaussian elimination is called *Gauss–Jordan elimination*
- it entails zeroing elements above the diagonal as well as below, transforming an $m \times n$ matrix into *reduced row echelon form*, i.e., a form where all pivoting entries in U are 1 and all entries above the pivots are zeros
- this is what you probably learnt in your undergraduate linear algebra class, e.g.,

$$A = \begin{bmatrix} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 11 & 5 & 35 \end{bmatrix} \to \begin{bmatrix} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 2 & 2 & 8 \end{bmatrix} \to \begin{bmatrix} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 0 & 0 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & -2 & -3 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix} = U$$

- the main drawback is that the elimination process can be numerically unstable, since the multipliers can be large
- furthermore the way it is done in undergraduate linear algebra courses is that the elimination matrices (i.e., the L and Π) are not stored

4. CONDENSED LU FACTORIZATION

- just like QR and SVD, LU factorization with complete pivoting has a condensed form too
- let $A \in \mathbb{R}^{m\times n}$ and $\operatorname{rank}(A) = r \le \min\{m, n\}$, recall that GECP yields
$$\begin{aligned}\Pi_1 A \Pi_2 &= LU \\ &= \begin{bmatrix} L_{11} & 0 \\ L_{21} & I_{m-r}\end{bmatrix}\begin{bmatrix} U_{11} & U_{12} \\ 0 & 0\end{bmatrix} \\ &= \begin{bmatrix} L_{11} \\ L_{21}\end{bmatrix}\begin{bmatrix} U_{11} & U_{12}\end{bmatrix} =: \widetilde{L}\widetilde{U}\end{aligned}$$
where $L_{11} \in \mathbb{R}^{r\times r}$ is unit lower triangular (thus nonsingular) and $U_{11} \in \mathbb{R}^{r\times r}$ is also nonsingular
- note that $\widetilde{L} \in \mathbb{R}^{m\times r}$ and $\widetilde{U} \in \mathbb{R}^{r\times n}$ and so
$$A = (\Pi_1^{\mathsf{T}}\widetilde{L})(\widetilde{U}\Pi_2^{\mathsf{T}})$$
is a rank-retaining factorization

5. LDU AND LDL^{T} FACTORIZATIONS

- if $A \in \mathbb{R}^{n\times n}$ has nonsingular principal submatrices $A_{1:k,1:k}$ for $k = 1, \dots, n$, then there exists a unit lower triangular matrix $L \in \mathbb{R}^{n\times n}$, a unit upper triangular matrix $U \in \mathbb{R}^{n\times n}$, and a diagonal matrix $D = \operatorname{diag}(d_{11}, \dots, d_{nn}) \in \mathbb{R}^{n\times n}$ such that
$$A = LDU = \begin{bmatrix} 1 & & & 0 \\ \ell_{21} & 1 & & \\ \vdots & & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & 1\end{bmatrix}\begin{bmatrix} d_{11} & & & \\ & d_{22} & & \\ & & \ddots & \\ & & & d_{nn}\end{bmatrix}\begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ & 1 & & u_{2n} \\ & & \ddots & \vdots \\ & & & 1\end{bmatrix}$$
- this is called the *LDU factorization* of A
- if A is furthermore symmetric, then $L = U^{\mathsf{T}}$ and this called the *LDL^{T} factorization*
- if they exist, then both LDU and LDL^{T} factorizations are unique (exercise)
- if a symmetric A has an LDL^{T} factorization and if $d_{ii} > 0$ for all $i = 1, \dots, n$, then A is positive definite
- in fact, even though $d_{11}, \dots, d_{nn}$ are not the eigenvalues of A (why not?), they must have the same signs as the eigenvalues of A, i.e., if A has p positive eigenvalues, q negative eigenvalues, and z zero eigenvalues, then there are exactly p, q, and z positive, negative, and zero entries in $d_{11}, \dots, d_{nn}$ — a consequence of the Sylvester law of inertia
- unfortunately, both LDU and LDL^{T} factorizations are difficult to compute because
  - the condition on the principal submatrices is difficult to check in advance
  - algorithms for computing them are invariably unstable because size of multipliers cannot be bounded in terms of the entries of A
- for example, the LDL^{T} factorization of a 2×2 symmetric matrix is
$$\begin{bmatrix} a & c \\ c & d\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ c/a & 1\end{bmatrix}\begin{bmatrix} a & c \\ 0 & d-(c/a)c\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ c/a & 1\end{bmatrix}\begin{bmatrix} a & 0 \\ 0 & d-(c/a)c\end{bmatrix}\begin{bmatrix} 1 & c/a \\ 0 & 1\end{bmatrix}$$
- so
$$\begin{bmatrix} \varepsilon & 1 \\ 1 & \varepsilon\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1/\varepsilon & 1\end{bmatrix}\begin{bmatrix} \varepsilon & 0 \\ 0 & \varepsilon - 1/\varepsilon\end{bmatrix}\begin{bmatrix} 1 & 1/\varepsilon \\ 0 & 1\end{bmatrix}$$
the elements of L and D are arbitrarily large when $|\varepsilon|$ is small
- note that you can't do partial or complete pivoting in LDL^{T} factorization since those could destroy the symmetry in A

- nonetheless there is one special case when LDL^{T} factorization not only exists but can be computed in an efficient and stable way — when A is positive definite

6. Positive definite matrices

- a symmetric matrix $A \in \mathbb{R}^{n\times n}$ is *positive definite* if $\mathbf{x}^{\mathsf{T}} A\mathbf{x} > 0$ for all nonzero $\mathbf{x}$
- a symmetric positive definite matrix has real and positive eigenvalues, and its leading principal submatrices all have positive determinants
- from the definition, it is easy to see that all diagonal elements are positive
- to solve the system $A\mathbf{x} = \mathbf{b}$ where A is symmetric positive definite, we can compute the *Cholesky factorization*
$$A = R^{\mathsf{T}} R$$
where R is upper triangular
- this factorization exists if and only if A is symmetric positive definite
- in fact, attempting to compute the Cholesky factorization of A is an efficient method for checking whether A is symmetric positive definite
- it is important to distinguish the Cholesky factorization from the *square root factorization*
- a square root of a matrix A is defined as a matrix S such that
$$S^2 = SS = A$$
- we often write $A^{-1/2}$ for S
- note that the matrix R in $A = R^{\mathsf{T}} R$ is not the square root of A, since it does not hold that $R^2 = A$ unless A is a diagonal matrix
- a symmetric square root of a symmetric positive definite A can be computed by using the fact that A has an eigendecomposition $A = Q\Lambda Q^{\mathsf{T}}$ where Λ is a diagonal matrix whose diagonal elements are the positive eigenvalues of A and Q is an orthogonal matrix whose columns are the eigenvectors of A
- it follows that
$$A = Q\Lambda Q^{\mathsf{T}} = (Q\Lambda^{1/2} Q^{\mathsf{T}})(Q\Lambda^{1/2} Q^{\mathsf{T}}) = SS$$
and so $S = Q\Lambda^{1/2} Q^{\mathsf{T}}$ is a square root of A, note that S is symmetric

7. Cholesky factorization

- the Cholesky factorization can be computed directly from the matrix equation $A = R^{\mathsf{T}} R$ where R is upper-triangular, much like how we derived Gram–Schmidt
- while it is conventional to write Cholesky factorization in the form $A = R^{\mathsf{T}} R$, it will be more natural later when we discuss the vectorized version of the algorithm to write $F = R^{\mathsf{T}}$ and $A = FF^{\mathsf{T}}$
- we can derive the algorithm for computing F by examining the matrix equation $A = R^{\mathsf{T}} R = FF^{\mathsf{T}}$ on an element-by-element basis, writing
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} f_{11} & & & \\ f_{21} & f_{22} & & \\ \vdots & \vdots & \ddots & \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix} \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{n1} \\ & f_{22} & & f_{n2} \\ & & \ddots & \vdots \\ & & & f_{nn} \end{bmatrix}$$
- from the above matrix multiplication we see that $f_{11}^2 = a_{11}$, from which it follows that
$$f_{11} = \sqrt{a_{11}}$$
- from the relationship $f_{11} f_{i1} = a_{1i}$ and the fact that we already know f_{11}, we obtain
$$f_{i1} = \frac{a_{1i}}{f_{11}}, \quad i = 2, \ldots, n$$

- proceeding to the second row of F, we see that $f_{21}^2 + f_{22}^2 = a_{22}$
- since we already know f_{21}, we have

$$f_{22} = \sqrt{a_{22} - f_{21}^2}$$

- if you know the fact that a positive definite matrix must have positive leading principal minors,[1] then you could deduce the term above in the square root is positive by examining the 2×2 principal minor:

$$a_{11}a_{22} - a_{12}^2 > 0$$

  and therefore

$$a_{22} > \frac{a_{12}^2}{a_{11}} = f_{21}^2$$

- next, we use the relation $f_{21}f_{i1} + f_{22}f_{i2} = a_{2i}$ to compute

$$f_{i2} = \frac{a_{2i} - f_{21}f_{i1}}{f_{22}}$$

- hence we get

$$\begin{aligned} a_{11} &= f_{11}^2, & \\ a_{i1} &= f_{11}f_{i1}, & i = 2, \dots, n \\ &\ \vdots & \\ a_{kk} &= f_{k1}^2 + f_{k2}^2 + \cdots + f_{kk}^2, & \\ a_{ik} &= f_{k1}f_{i1} + \cdots + f_{kk}f_{ik}, & i = k+1, \dots, n \end{aligned}$$

- the resulting algorithm that runs for $k = 1, \dots, n$ is

$$\begin{aligned} f_{kk} &= \left(a_{kk} - \textstyle\sum_{j=1}^{k-1} f_{kj}^2\right)^{1/2}, & \\ f_{ik} &= \frac{\left(a_{ik} - \sum_{j=1}^{k-1} f_{kj}f_{ij}\right)}{f_{kk}}, & i = k+1, \dots, n \end{aligned}$$

- you could use induction to show that the term in the square root is always positive but we'll soon see a more elegant vectorized version showing that this algorithm doesn't ever require taking square roots of negative numbers
- this algorithm requires roughly half as many operations as Gaussian elimination
- note that

$$a_{kk} = f_{k1}^2 + f_{k2}^2 + \cdots + f_{kk}^2$$

  which implies that

$$|f_{ki}| \le \sqrt{a_{kk}}$$

- in other words, the entries of F are automatically bounded by the (square root of the) diagonal entries of A
- this is why there no need to do any pivoting for Cholesky factorization

[1] If you don't, see `https://en.wikipedia.org/wiki/Sylvester's_criterion`; now you do.

8. ANOTHER LOOK AT CHOLESKY

- instead of considering an elementwise algorithm, we can also derive a vectorized version
- this is analogous to our discussions of Householder QR and Gaussian elimination for LU
- let $F = [\mathbf{f}_1, \dots, \mathbf{f}_n]$ where $\mathbf{f}_i$ is the ith column of the lower-triangular matrix F so
$$A = FF^{\mathsf{T}} = \mathbf{f}_1\mathbf{f}_1^{\mathsf{T}} + \cdots + \mathbf{f}_n\mathbf{f}_n^{\mathsf{T}}$$
- we start by observing that
$$\mathbf{f}_1 = \frac{1}{\sqrt{a_{11}}}\mathbf{a}_1$$
where $\mathbf{a}_i$ is the ith column of A
- then we compute
$$A^{(2)} = A - \mathbf{f}_1\mathbf{f}_1^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & A_2 & \\ 0 & & & \end{bmatrix}$$
- note that
$$A = B\begin{bmatrix} 1 & 0 \\ 0 & A_2 \end{bmatrix}B^{\mathsf{T}}$$
where B is the identity matrix with its first column replaced by $\mathbf{f}_1$
$$B = [\mathbf{f}_1, \mathbf{e}_2, \dots, \mathbf{e}_n] = \begin{bmatrix} f_{11} & & & \\ f_{21} & 1 & & \\ \vdots & & \ddots & \\ f_{n1} & & & 1 \end{bmatrix}$$
- it follows that A_2 is positive definite since
$$\begin{bmatrix} 1 & 0 \\ 0 & A_2 \end{bmatrix} = B^{-1}AB^{-\mathsf{T}}$$
is positive definite:
$$\mathbf{x}^{\mathsf{T}}A_2\mathbf{x} = \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix}^{\mathsf{T}}\begin{bmatrix} 1 & 0 \\ 0 & A_2 \end{bmatrix}\begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} = (B^{-\mathsf{T}}\mathbf{y})^{\mathsf{T}}A(B^{-}{}_{\mathsf{T}}\mathbf{y}) > 0$$
for all $\mathbf{x} \neq \mathbf{0}$ (or if you know Sylvester law of inertia, you can apply it to deduce the same thing since C is lower triangular)
- so we may repeat the process on A_2
- we partition the matrix A_2 into columns, writing $A_2 = \begin{bmatrix} \mathbf{a}_2^{(2)} & \mathbf{a}_3^{(2)} & \cdots & \mathbf{a}_n^{(2)} \end{bmatrix}$ and then compute
$$\mathbf{f}_2 = \frac{1}{\sqrt{a_{22}^{(2)}}}\begin{bmatrix} 0 \\ \mathbf{a}_2^{(2)} \end{bmatrix}$$
- we then compute
$$A^{(3)} = A^{(2)} - \mathbf{f}_2\mathbf{f}_2^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & A_3 & \\ 0 & 0 & & \end{bmatrix}$$
and so on

9. some observations about Cholesky decomposition

- we also have the relationship

$$\det A = \det F \det F^{\mathsf{T}} = (\det F)^2 = f_{11}^2 f_{22}^2 \cdots f_{nn}^2$$

- is the Cholesky decomposition unique?
- employing a similar approach to the one used to prove the uniquess of the LU factorization, we assume that A has two Cholesky factorizations

$$A = F_1 F_1^{\mathsf{T}} = F_2 F_2^{\mathsf{T}}$$

- then

$$F_2^{-1} F_1 = F_2^{\mathsf{T}} F_1^{-\mathsf{T}}$$

but since F_1 and F_2 are lower triangular, both matrices must be diagonal
- let

$$F_2^{-1} F_1 = D = F_2^{\mathsf{T}} F_1^{-\mathsf{T}}$$

- so $F_1 = F_2 D$ and thus $F_1^{\mathsf{T}} = D F_2^{\mathsf{T}}$ and we get $D^{-1} = F_2^{\mathsf{T}} F_1^{-\mathsf{T}}$
- in other words, $D^{-1} = D$ or $D^2 = I$
- hence D must have diagonal elements equal to ± 1
- since we require that the diagonal elements be positive, it follows that the factorization is unique
- in computing the Cholesky factorization, no row interchanges are necessary because A is positive definite, so the number of operations required to compute F is approximately $n^3/3$
- a simple variant of the algorithm Cholesky factorization yields the LDL^{T} factorization

$$A = LDL^{\mathsf{T}}$$

where L is a unit lower triangular matrix, and D is a diagonal matrix with positive diagonal elements
- the algorithm is sometimes called the *square-root-free Cholesky factorization* since unlike in the usual Cholesky factorization, it does not require taking square roots (which can be expensive, most computer hardware and software use Newton–Raphson method to extract square roots)
- the LDL^{T} and Cholesky factorizations are related by

$$F = LD^{1/2}$$

- also the QR factorization of A and Cholesky factorization of $A^{\mathsf{T}}A$ are related by

$$A^{\mathsf{T}}A = R^{\mathsf{T}}Q^{\mathsf{T}}QR = R^{\mathsf{T}}R$$

10. costs of various matrix decompositions

- in modern computing, flop counts are pretty meaningless:
  `http://www.stat.uchicago.edu/~lekheng/courses/309/flops/`
- but it can still be a useful guide
- the following table summarizes flop counts of some standard matrix decompositions for $A \in \mathbb{C}^{m\times n}$
- $m = n$ for all except Cholesky and singular value decomposition, where $m \geq n$

decomposition	*algorithm*	*form*	*flops*
LU factorization	Gaussian elimination row pivoting	$PA = LU$	$2n^3/3$
Cholesky factorization	Cholesky algorithm	$A = R^*R$	$n^3/3$
QR factorization	Householder algorithm	$A = QR$	$2n^2(m - n/3)$ for R;
			$4(m^2n - mn^2 + n^3/3)$ for full Q;
			$2n^2(m - n/3)$ for condensed Q;
Singular value decompositiion	Golub–Reinsch algorithm	$A = U\Sigma V^*$	$14mn^2 + 8n^3$ for condensed form
Hessenberg decompositiion	Householder tridiagonalization	$A = QHQ^*$	$14n^3/3$
Tridiagonal decompositiion	Householder tridiagonalization	$A = QTQ^* = A^*$	$8n^3/3$
Schur decompositiion	Francis QR algorithm	$A = QRQ^*$	$25n^3$
Eigenvalue decompositiion	Francis QR algorithm	$A = Q\Lambda Q^* = A^*$	$9n^3$