Chapter 3

Gaussian Elimination, LU-Factorization, and Cholesky Factorization

3.1 Gaussian Elimination and LU-Factorization

Let A be an $n \times n$ matrix, let $b \in \mathbb{R}^n$ be an n-dimensional vector and assume that A is invertible.

Our goal is to solve the system $Ax = b$. Since A is assumed to be invertible, we know that this system has a unique solution, $x = A^{-1}b$.

Experience shows that two counter-intuitive facts are revealed:

(1) One should avoid computing the inverse, A^{-1}, of A explicitly. This is because this would amount to solving the n linear systems, $Au^{(j)} = e_j$, for $j = 1, \ldots, n$, where $e_j = (0, \ldots, 1, \ldots, 0)$ is the jth canonical basis vector of $\mathbb{R}^n$ (with a 1 is the jth slot).

By doing so, we would replace the resolution of a single system by the resolution of n systems, and we would still have to multiply A^{-1} by b.

(2) One does not solve (large) linear systems by computing determinants (using Cramer's formulae).

This is because this method requires a number of additions (resp. multiplications) proportional to $(n+1)!$ (resp. $(n+2)!$).

The key idea on which most direct methods are based is that if A is an *upper-triangular matrix*, which means that $a_{ij} = 0$ for $1 \leq j < i \leq n$ (resp. lower-triangular, which means that $a_{ij} = 0$ for $1 \leq i < j \leq n$), then computing the solution, x, is trivial.

Indeed, say A is an upper-triangular matrix

$$A = \begin{pmatrix} a_{1\,1} & a_{1\,2} & \cdots & a_{1\,n-2} & a_{1\,n-1} & a_{1\,n} \\ 0 & a_{2\,2} & \cdots & a_{2\,n-2} & a_{2\,n-1} & a_{2\,n} \\ 0 & 0 & \ddots & \vdots & \vdots & \vdots \\ & & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & a_{n-1\,n-1} & a_{n-1\,n} \\ 0 & 0 & \cdots & 0 & 0 & a_{n\,n} \end{pmatrix}.$$

Then, $\det(A) = a_{1\,1}a_{2\,2}\cdots a_{n\,n} \neq 0$, and we can solve the system $Ax = b$ from bottom-up by *back-substitution*, i.e., first we compute x_n from the last equation, next plug this value of x_n into the next to the last equation and compute x_{n-1} from it, etc.

This yields

$$\begin{aligned} x_n &= a_{n\,n}^{-1} b_n \\ x_{n-1} &= a_{n-1\,n-1}^{-1}(b_{n-1} - a_{n-1\,n} x_n) \\ &\vdots \\ x_1 &= a_{1\,1}^{-1}(b_1 - a_{1\,2}x_2 - \cdots - a_{1\,n}x_n). \end{aligned}$$

If A was lower-triangular, we would solve the system from top-down by *forward-substitution*.

Thus, what we need is a method for transforming a matrix to an equivalent one in upper-triangular form.

This can be done by *elimination*.

Consider the following example:

$$\begin{array}{rcrcrcr} 2x & + & y & + & z & = & 5 \\ 4x & - & 6y & & & = & -2 \\ -2x & + & 7y & + & 2z & = & 9. \end{array}$$

We can eliminate the variable x from the second and the third equation as follows: Subtract twice the first equation from the second and add the first equation to the third. We get the new system

$$\begin{array}{rcrcrcr} 2x & + & y & + & z & = & 5 \\ & - & 8y & - & 2z & = & -12 \\ & & 8y & + & 3z & = & 14. \end{array}$$

This time, we can eliminate the variable y from the third equation by adding the second equation to the third:

$$\begin{array}{rcrcrcr} 2x & + & y & + & z & = & 5 \\ & - & 8y & - & 2z & = & -12 \\ & & & & z & = & 2. \end{array}$$

This last system is upper-triangular.

Using back-substitution, we find the solution: $z = 2$, $y = 1$, $x = 1$.

Observe that we have performed only *row operations*.

The general method is to *iteratively eliminate variables* using simple row operations (namely, adding or subtracting a multiple of a row to another row of the matrix) while simultaneously applying these operations to the vector b, to obtain a system, $MAx = Mb$, where MA is *upper-triangular*.

Such a method is called *Gaussian elimination*.

However, one extra twist is needed for the method to work in all cases: It may be necessary to *permute rows*, as illustrated by the following example:

$$\begin{aligned} x + y + z &= 1 \\ x + y + 3z &= 1 \\ 2x + 5y + 8z &= 1. \end{aligned}$$

In order to eliminate x from the second and third row, we subtract the first row from the second and we subtract twice the first row from the third:

$$\begin{aligned} x + y + z &= 1 \\ 2z &= 0 \\ 3y + 6z &= -1. \end{aligned}$$

Now, the trouble is that y does not occur in the second row; so, we can't eliminate y from the third row by adding or subtracting a multiple of the second row to it.

The remedy is simple: *permute* the second and the third row! We get the system:

$$\begin{aligned} x + y + z &= 1 \\ 3y + 6z &= -1 \\ 2z &= 0, \end{aligned}$$

which is already in triangular form.

Another example where some permutations are needed is:

$$\begin{aligned} z &= 1 \\ -2x + 7y + 2z &= 1 \\ 4x - 6y &= -1. \end{aligned}$$

First, we *permute* the first and the second row, obtaining

$$\begin{aligned} -2x + 7y + 2z &= 1 \\ z &= 1 \\ 4x - 6y &= -1, \end{aligned}$$

and then, we add twice the first row to the third (to eliminate x) obtaining:

$$\begin{aligned} -2x + 7y + 2z &= 1 \\ z &= 1 \\ 8y + 4z &= 1. \end{aligned}$$

Again, we permute the second and the third row, getting

$$\begin{aligned} -2x + 7y + 2z &= 1 \\ 8y + 4z &= 1 \\ z &= 1, \end{aligned}$$

an upper-triangular system.

Of course, in this example, z is already solved and we could have eliminated it first, but for the general method, we need to proceed in a systematic fashion.

We now describe the method of *Gaussian Elimination* applied to a linear system, $Ax = b$, where A is assumed to be invertible.

We use the variable k to keep track of the stages of elimination. Initially, $k = 1$.

(1) The first step is to *pick some nonzero entry, $a_{i\,1}$, in the first column of A*. Such an entry must exist, since A is invertible (otherwise, we would have $\det(A) = 0$). The actual choice of such an element has some impact on the numerical stability of the method, but this will be examined later. For the time being, we assume that some arbitrary choice is made. This chosen element is called the *pivot* of the elimination step and is denoted π_1 (so, in this first step, $\pi_1 = a_{i\,1}$).

(2) Next, we *permute* the row (i) corresponding to the pivot with the first row. Such a step is called *pivoting*. So, after this permutation, the first element of the first row is nonzero.

(3) We now *eliminate* the variable x_1 from all rows except the first by adding suitable multiples of the first row to these rows. More precisely we add $-a_{i\,1}/\pi_1$ times the first row to the ith row, for $i = 2, \ldots, n$. At the end of this step, all entries in the first column are zero except the first.

(4) Increment k by 1. If $k = n$, stop. Otherwise, $k < n$, and then iteratively *repeat* steps (1), (2), (3) on the $(n-k+1) \times (n-k+1)$ subsystem obtained by deleting the first $k-1$ rows and $k-1$ columns from the current system.

If we let $A_1 = A$ and $A_k = (a^k_{i\,j})$ be the matrix obtained after $k-1$ elimination steps ($2 \leq k \leq n$), then the kth elimination step is applied to the matrix A_k of the form

$$A_k = \begin{pmatrix} a^k_{1\,1} & a^k_{1\,2} & \cdots & \cdots & \cdots & a^k_{1\,n} \\ & a^k_{2\,2} & \cdots & \cdots & \cdots & a^k_{2\,n} \\ & & \ddots & \vdots & & \vdots \\ & & & a^k_{k\,k} & \cdots & a^k_{k\,n} \\ & & & \vdots & & \vdots \\ & & & a^k_{n\,k} & \cdots & a^k_{n\,n} \end{pmatrix}.$$

Actually, note

$$a^k_{i\,j} = a^i_{i\,j}$$

for all i, j with $1 \leq i \leq k-1$ and $i \leq j \leq n$, since the first $k-1$ rows remain unchanged after the $(k-1)$th step.

Now, we will prove later that $\det(A_k) = \pm \det(A)$.

Since A is invertible, some entry $a^k_{i\,k}$ with $k \leq i \leq n$ is nonzero; so, one of these entries can be chosen as pivot, and we permute the kth row with the ith row, obtaining the matrix $\alpha^k = (\alpha^k_{j\,l})$.

The new pivot is $\pi_k = \alpha^k_{k\,k}$, and we zero the entries $i = k+1, \ldots, n$ in column k by adding $-\alpha^k_{i\,k}/\pi_k$ times row k to row i. At the end of this step, we have A_{k+1}.

Observe that the first $k-1$ rows of A_k are identical to the first $k-1$ rows of A_{k+1}.

The permutation of the kth row with the ith row is achieved by multiplying A on the left by the *transposition matrix* $P(i,k)$, which is the matrix obtained from the identity matrix by permuting rows i and k, i.e.,

$$P(i,k) = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 0 & & & & 1 & & \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 1 & & & \\ & & 1 & & & & 0 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{pmatrix}.$$

Observe that $\det(P(i,k)) = -1$.

Therefore, during the permutation step (2), if row k and row i need to be permuted, the matrix A is multiplied on the left by the matrix P_k such that $P_k = P(i,k)$, else we set $P_k = I$.

Adding β times row j to row i is achieved by multiplying A on the left by the *elementary matrix*,

$$E_{i,j;\beta} = I + \beta e_{i,j},$$

where

$$(e_{i,j})_{k\,l} = \begin{cases} 1 & \text{if } k = i \text{ and } l = j \\ 0 & \text{if } k \neq i \text{ or } l \neq j, \end{cases}$$

i.e.,

$$E_{i,j;\beta} = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 1 & & & \\ & & \beta & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{pmatrix}.$$

Observe that the inverse of $E_{i,j;\beta} = I + \beta e_{i\,j}$ is $E_{i,j;-\beta} = I - \beta e_{i,j}$ and that $\det(E_{i,j;\beta}) = 1$.

Therefore, during step 3 (the elimination step), the matrix A is multiplied on the left by a product, E_k, of matrices of the form $E_{i,k;\beta_{i,k}}$.

Consequently, we see that

$$A_{k+1} = E_k P_k A_k.$$

The fact that $\det(P(i,k)) = -1$ and that $\det(E_{i,j;\beta}) = 1$ implies immediately the fact claimed above: We always have $\det(A_k) = \pm \det(A)$. Furthermore, since

$$A_{k+1} = E_k P_k A_k$$

and since Gaussian elimination stops for $k = n$, the matrix

$$A_n = E_{n-1} P_{n-1} \cdots E_2 P_2 E_1 P_1 A$$

is *upper-triangular*.

Also note that if we let

$$M = E_{n-1} P_{n-1} \cdots E_2 P_2 E_1 P_1,$$

then $\det(M) = \pm 1$, and

$$\det(A) = \pm \det(A_n).$$

We can summarize all this in the following theorem:

Theorem 3.1. *(Gaussian Elimination) Let A be an $n \times n$ matrix (invertible or not). Then there is some invertible matrix, M, so that $U = MA$ is upper-triangular. The pivots are all nonzero iff A is invertible.*

Remark: Obviously, the matrix M can be computed as

$$M = E_{n-1}P_{n-1} \cdots E_2P_2E_1P_1,$$

but this expression is of no use.

Indeed, what we need is M^{-1}; when no permutations are needed, it turns out that M^{-1} can be obtained immediately from the matrices E_k's, in fact, from their inverses, and no multiplications are necessary.

Remark: Instead of looking for an invertible matrix, M, so that MA is upper-triangular, we can look for an invertible matrix, M, so that *MA is a diagonal matrix*.

Only a simple change to Gaussian elimination is needed.

At every stage, k, after the pivot has been found and pivoting been performed, if necessary, in addition to adding suitable multiples of the kth row to the rows *below* row k in order to zero the entries in column k for $i = k + 1, \ldots, n$, also add suitable multiples of the kth row to the rows *above* row k in order to zero the entries in column k for $i = 1, \ldots, k - 1$.

Such steps are also achieved by multiplying on the left by elementary matrices $E_{i,k;\beta_{i,k}}$, except that $i < k$, so that these matrices are not lower-diagonal matrices.

Nevertheless, at the end of the process, we find that
$A_n = MA$, is a diagonal matrix.

This method is called the *Gauss-Jordan factorization*. Because it is more expansive than Gaussian elimination, this method is not used much in practice.

However, Gauss-Jordan factorization can be used to compute the inverse of a matrix, A.

It remains to discuss the choice of the pivot, and also conditions that guarantee that no permutations are needed during the Gaussian elimination process.

We begin by stating a necessary and sufficient condition for an invertible matrix to have an LU-factorization (i.e., Gaussian elimination does not require pivoting).

We say that an invertible matrix, A, has an *LU-factorization* if it can be written as $A = LU$, where U is *upper-triangular* invertible and L is *lower-triangular*, with $L_{i\,i} = 1$ for $i = 1, \ldots, n$.

A lower-triangular matrix with diagonal entries equal to 1 is called a *unit lower-triangular* matrix.

Given an $n \times n$ matrix, $A = (a_{i\,j})$, for any k, with $1 \leq k \leq n$, let $A[1..k, 1..k]$ denote the submatrix of A whose entries are $a_{i\,j}$, where $1 \leq i, j \leq k$.

Proposition 3.2. *Let A be an invertible $n \times n$-matrix. Then, A, has an LU-factorization, $A = LU$, iff every matrix $A[1..k, 1..k]$ is invertible for $k = 1, \ldots, n$.*

Corollary 3.3. *(LU-Factorization) Let A be an invertible $n \times n$-matrix. If every matrix $A[1..k, 1..k]$ is invertible for $k = 1, \ldots, n$, then Gaussian elimination requires no pivoting and yields an LU-factorization, $A = LU$.*

The reader should verify that the example below is indeed an LU-factorization.

$$\begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

One of the main reasons why the existence of an LU-factorization for a matrix, A, is interesting is that if we need to solve *several* linear systems, $Ax = b$, corresponding to the same matrix, A, we can do this cheaply by solving the two triangular systems

$$Lw = b, \quad \text{and} \quad Ux = w.$$

As we will see a bit later, symmetric positive definite matrices satisfy the condition of Proposition 3.2.

Therefore, linear systems involving symmetric positive definite matrices can be solved by Gaussian elimination without pivoting.

Actually, it is possible to do better: This is the Cholesky factorization.

The following easy proposition shows that, in principle, A can be premultipied by some permutation matrix, P, so that PA can be converted to upper-triangular form without using any pivoting.

Proposition 3.4. *Let A be an invertible $n \times n$-matrix. Then, there is some permutation matrix, P, so that $PA[1..k, 1..k]$ is invertible for $k = 1, \ldots, n$.*

Remark: One can also prove Proposition 3.4 using a clever reordering of the Gaussian elimination steps.

Theorem 3.5. *For every invertible $n \times n$-matrix, A, there is some permutation matrix, P, some upper-triangular matrix, U, and some unit lower-triangular matrix, L, so that $PA = LU$ (recall, $L_{i\,i} = 1$ for $i = 1, \ldots, n$). Furthermore, if $P = I$, then L and U are unique and they are produced as a result of Gaussian elimination without pivoting. Furthermore, if $P = I$, then L is simply obtained from the E_k^{-1}'s.*

Remark: It can be shown that Gaussian elimination + back-substitution requires $n^3/3 + O(n^2)$ additions, $n^3/3 + O(n^2)$ multiplications and $n^2/2 + O(n)$ divisions.

Let us now briefly comment on the choice of a pivot.

Although theoretically, any pivot can be chosen, the possibility of roundoff errors implies that it is *not a good idea to pick very small pivots*. The following example illustrates this point.

$$\begin{aligned} 10^{-4}x \ + \ y &= 1 \\ x \ + \ y &= 2. \end{aligned}$$

Since 10^{-4} is nonzero, it can be taken as pivot, and we get

$$\begin{aligned} 10^{-4}x \ + \qquad y &= 1 \\ (1 - 10^4)y &= 2 - 10^4. \end{aligned}$$

Thus, the exact solution is

$$x = \frac{10^4}{10^4 - 1}, \quad y = \frac{10^4 - 2}{10^4 - 1}.$$

However, if roundoff takes place on the fourth digit, then $10^4 - 1 = 9999$ and $10^4 - 2 = 9998$ will be rounded off both to 9990, and then, the solution is $x = 0$ and $y = 1$, very far from the exact solution where $x \approx 1$ and $y \approx 1$.

The problem is that we picked a *very small pivot*.

If instead we permute the equations, the pivot is 1, and after elimination, we get the system

$$\begin{aligned} x \; + \qquad\quad y \;&=\; 2 \\ (1 - 10^{-4})y \;&=\; 1 - 2 \times 10^{-4}. \end{aligned}$$

This time, $1 - 10^{-4} = -0.9999$ and $1 - 2 \times 10^{-4} = -0.9998$ are rounded off to 0.999 and the solution is $x = 1, y = 1$, much closer to the exact solution.

To remedy this problem, one may use the strategy of *partial pivoting*.

This consists of choosing during step k ($1 \leq k \leq n-1$) one of the entries $a^k_{i\,k}$ such that

$$|a^k_{i\,k}| = \max_{k \leq p \leq n} |a^k_{p\,k}|.$$

By maximizing the value of the pivot, we avoid dividing by undesirably small pivots.

Remark: A matrix, A, is called *strictly column diagonally dominant* iff

$$|a_{j\,j}| > \sum_{i=1,\, i \neq j}^{n} |a_{i\,j}|, \quad \text{for } j = 1, \ldots, n$$

(resp. *strictly row diagonally dominant* iff

$$|a_{i\,i}| > \sum_{j=1,\, j \neq i}^{n} |a_{i\,j}|, \quad \text{for } i = 1, \ldots, n.)$$

It has been known for a long time (before 1900, say by Hadamard) that if a matrix, A, is strictly column diagonally dominant (resp. strictly row diagonally dominant), then it is invertible. (This is a good exercise, try it!)

It can also be shown that if A is strictly column diagonally dominant, then Gaussian elimination with partial pivoting does not actually require pivoting.

Another strategy, called *complete pivoting*, consists in choosing some entry $a^k_{i\,j}$, where $k \leq i, j \leq n$, such that

$$|a^k_{i\,j}| = \max_{k \leq p, q \leq n} |a^k_{p\,q}|.$$

However, in this method, if the chosen pivot is not in column k, it is also necessary to *permute columns*.

This is achieved by multiplying on the right by a permutation matrix.

However, complete pivoting tends to be too expansive in practice, and partial pivoting is the method of choice.

A special case where the LU-factorization is particularly efficient is the case of tridiagonal matrices, which we now consider.

3.2 Gaussian Elimination of Tridiagonal Matrices

Consider the tridiagonal matrix

$$A = \begin{pmatrix} b_1 & c_1 & & & & & \\ a_2 & b_2 & c_2 & & & & \\ & a_3 & b_3 & c_3 & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_{n-2} & b_{n-2} & c_{n-2} & & \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{pmatrix}.$$

Define the sequence

$$\begin{aligned} \delta_0 &= 1, \\ \delta_1 &= b_1, \\ \delta_k &= b_k\delta_{k-1} - a_k c_{k-1}\delta_{k-2}, \quad 2 \leq k \leq n. \end{aligned}$$

Proposition 3.6. *If A is the tridiagonal matrix above, then $\delta_k = \det(A[1..k, 1..k])$, for $k = 1, \ldots, n$.*

Theorem 3.7. *If A is the tridiagonal matrix above and $\delta_k \neq 0$ for $k = 1, \ldots, n$, then A has the following LU-factorization:*

$$A = \begin{pmatrix} 1 & & & & & \\ a_2\frac{\delta_0}{\delta_1} & 1 & & & & \\ & a_3\frac{\delta_1}{\delta_2} & 1 & & & \\ & & \ddots & \ddots & & \\ & & & a_{n-1}\frac{\delta_{n-3}}{\delta_{n-2}} & 1 & \\ & & & & a_n\frac{\delta_{n-2}}{\delta_{n-1}} & 1 \end{pmatrix} \begin{pmatrix} \frac{\delta_1}{\delta_0} & c_1 & & & & \\ & \frac{\delta_2}{\delta_1} & c_2 & & & \\ & & \frac{\delta_3}{\delta_2} & c_3 & & \\ & & & \ddots & \ddots & \\ & & & & \frac{\delta_{n-1}}{\delta_{n-2}} & c_{n-1} \\ & & & & & \frac{\delta_n}{\delta_{n-1}} \end{pmatrix}.$$

It follows that there is a simple method to solve a linear system, $Ax = d$, where A is tridiagonal (and $\delta_k \neq 0$ for $k = 1, \ldots, n$).

For this, it is convenient to "squeeze" the diagonal matrix, Δ, defined such that $\Delta_{k\,k} = \delta_k/\delta_{k-1}$, into the factorization so that $A = (L\Delta)(\Delta^{-1}U)$, and if we let

$$\begin{aligned} z_1 &= \frac{c_1}{b_1}, \\ z_k &= c_k \, \frac{\delta_{k-1}}{\delta_k}, \quad 2 \leq k \leq n-1, \\ z_n &= \frac{\delta_n}{\delta_{n-1}} = b_n - a_n z_{n-1}, \end{aligned}$$

$A = (L\Delta)(\Delta^{-1}U)$ is written as

$$A = \begin{pmatrix} \frac{c_1}{z_1} & & & & & \\ a_2 & \frac{c_2}{z_2} & & & & \\ & a_3 & \frac{c_3}{z_3} & & & \\ & & \ddots & \ddots & & \\ & & & a_{n-1} & \frac{c_{n-1}}{z_{n-1}} & \\ & & & & a_n & z_n \end{pmatrix} \begin{pmatrix} 1 & z_1 & & & & & \\ & 1 & z_2 & & & & \\ & & 1 & z_3 & & & \\ & & & \ddots & \ddots & & \\ & & & & 1 & z_{n-2} & \\ & & & & & 1 & z_{n-1} \\ & & & & & & 1 \end{pmatrix}.$$

As a consequence, the system $Ax = d$ can be solved by constructing three sequences: First, the sequence

$$\begin{aligned} z_1 &= \frac{c_1}{b_1}, \\ z_k &= \frac{c_k}{b_k - a_k z_{k-1}}, \quad k = 2, \ldots, n-1, \\ z_n &= b_n - a_n z_{n-1}, \end{aligned}$$

corresponding to the recurrence $\delta_k = b_k\delta_{k-1} - a_k c_{k-1}\delta_{k-2}$ and obtained by dividing both sides of this equation by δ_{k-1}, next

$$w_1 = \frac{d_1}{b_1}, \quad w_k = \frac{d_k - a_k w_{k-1}}{b_k - a_k z_{k-1}}, \quad k = 2, \ldots, n,$$

corresponding to solving the system $L\Delta w = d$, and finally

$$x_n = w_n, \quad x_k = w_k - z_k x_{k+1}, \quad k = n-1, n-2, \ldots, 1,$$

corresponding to solving the system $\Delta^{-1}Ux = w$.

Remark: It can be verified that this requires $3(n-1)$ additions, $3(n-1)$ multiplications, and $2n$ divisions, a total of $8n-6$ operations, which is much less that the $O(2n^3/3)$ required by Gaussian elimination in general.

We now consider the special case of symmetric positive definite matrices (SPD matrices).

Recall that an $n \times n$ symmetric matrix, A, is *positive definite* iff

$$x^\top Ax > 0 \quad \text{for all } x \in \mathbb{R}^n \text{ with } x \neq 0.$$

Equivalently, A is symmetric positive definite iff all its eigenvalues are strictly positive.

The following facts about a symmetric positive definite matrice, A, are easily established:

(1) The matrix A is invertible. (Indeed, if $Ax = 0$, then $x^\top Ax = 0$, which implies $x = 0$.)

(2) We have $a_{i\,i} > 0$ for $i = 1, \ldots, n$. (Just observe that for $x = e_i$, the ith canonical basis vector of $\mathbb{R}^n$, we have $e_i^\top Ae_i = a_{i\,i} > 0$.)

(3) For every $n \times n$ invertible matrix, Z, the matrix $Z^\top AZ$ is symmetric positive definite iff A is symmetric positive definite.

Next, we prove that a symmetric positive definite matrix has a special LU-factorization of the form $A = BB^\top$, where B is a lower-triangular matrix whose diagonal elements are strictly positive.

This is the *Cholesky factorization*.

3.3 SPD Matrices and the Cholesky Decomposition

First, we note that a symmetric positive definite matrix satisfies the condition of Proposition 3.2.

Proposition 3.8. *If A is a symmetric positive definite matrix, then $A[1..k, 1..k]$ is invertible for $k = 1, \ldots, n$.*

Let A be a symmetric positive definite matrix and write

$$A = \begin{pmatrix} a_{1\,1} & W^\top \\ W & B \end{pmatrix}.$$

Since A is symmetric positive definite, $a_{1\,1} > 0$, and we can compute $\alpha = \sqrt{a_{1\,1}}$. The trick is that we can factor A uniquely as

$$\begin{aligned} A &= \begin{pmatrix} a_{1\,1} & W^\top \\ W & B \end{pmatrix} \\ &= \begin{pmatrix} \alpha & 0 \\ W/\alpha & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & B - WW^\top/a_{1\,1} \end{pmatrix} \begin{pmatrix} \alpha & W^\top/\alpha \\ 0 & I \end{pmatrix}, \end{aligned}$$

i.e., as $A = B_1 A_1 B_1^\top$, where B_1 is lower-triangular with positive diagonal entries.

Thus, B_1 is invertible, and by fact (3) above, A_1 is also symmetric positive definite.

Theorem 3.9. *(Cholesky Factorization) Let A be a symmetric positive definite matrix. Then, there is some lower-triangular matrix, B, so that $A = BB^\top$. Furthermore, B can be chosen so that its diagonal elements are strictly positive, in which case, B is unique.*

Remark: If $A = BB^\top$, where B is any invertible matrix, then A is symmetric positive definite.

The proof of Theorem 3.9 immediately yields an algorithm to compute B from A. For $j = 1, \ldots, n$,

$$b_{j\,j} = \left(a_{j\,j} - \sum_{k=1}^{j-1} b_{j\,k}^2 \right)^{1/2},$$

and for $i = j + 1, \ldots, n$,

$$b_{i\,j} = \left(a_{i\,j} - \sum_{k=1}^{j-1} b_{i\,k} b_{j\,k} \right) / b_{j\,j}.$$

The Cholesky factorization can be used to solve linear systems, $Ax = b$, where A is symmetric positive definite:

Solve the two systems $Bw = b$ and $B^{\top}x = w$.

Remark: It can be shown that this methods requires $n^3/6 + O(n^2)$ additions, $n^3/6 + O(n^2)$ multiplications, $n^2/2 + O(n)$ divisions, and $O(n)$ square root extractions.

Thus, the Cholesky method requires half of the number of operations required by Gaussian elimination (since Gaussian elimination requires $n^3/3 + O(n^2)$ additions, $n^3/3 + O(n^2)$ multiplications, and $n^2/2 + O(n)$ divisions).

It also requires half of the space (only B is needed, as opposed to both L and U).

Furthermore, it can be shown that Cholesky's method is numerically stable.

Remark: Proposition 3.8 can be strengthened as follows: *A symmetric matrix is positive definite iff* $\det(A[1..k, 1..k]) > 0$ *for* $k = 1, \ldots, n$.

The above fact is known as *Sylvester's criterion*.

Another criterion is that Gaussian elimination needs no pivoting and that all the pivots are strictly positive.

For more on the stability analysis and efficient implementation methods of Gaussian elimination, LU-factoring and Cholesky factoring, see Demmel [11], Trefethen and Bau [30], Ciarlet [9], Golub and Van Loan [15], Strang [27, 28], and Kincaid and Cheney [19].