
0. Linear Algebra. It is assumed that everyone has taken or at least knows the material of a first-semester linear algebra course equivalent to Mathematics 250 at Rutgers–NB. Most of Atkinson’s §7.1, pp. 463–471, and similarly of Burden & Faires’ §§6.1, 6.3 and 6.4, represents nothing more than a review of parts of the material of such a course.

1. Solving Linear Systems by Gaussian Elimination and Back-substitution. Atkinson deals with the basic (and essentially hand-)computational aspects of this technique in the first three pages of §8.1, pp. 508–510; Burden & Faires deal with it in their §§6.1 and 6.2; Van Loan gives the $3 \times 3$ case a light dusting in his §6.3.1. The method can be described in a few words, as follows. Imagine the equations of the system written one above the other. Multiply the first equation by an appropriate constant and subtract it from the second equation so that the coefficient of $x_1$ in the new second equation is zero; repeat this process with the third, fourth, etc., equations, working downward, until only the first equation has an $x_1$ term (with nonzero coefficient). Leave the first equation alone from now on; repeat the process just described, working downward with the “new” second equation (the second equation in the original system will probably have been altered by subtracting a multiple of the first equation), until only it has an $x_2$ term. Leave the second equation alone from now on; repeat the process just described . . . . At the $(k + 1)$-st trip around this loop, the only part of the matrix of coefficients in which one is not just juggling zeros is the $(n - k) \times (n - k)$ submatrix containing the diagonal from $a_{k+1,k+1}$ to $a_{n,n}$ inclusive. The result at the end of the process will be an “upper-triangular” system whose solution vector is the same as that of the original system. Such a system can be solved “from the bottom up,” whence the name back-solution, as follows. The last equation has the form $b^{(n)}_n = b^{(n)}_n$, so it tells you the value of $x_n$. With the value of $x_n$ available, the next-to-last equation, which only involves $x_{n-1}$ and $x_n$, can be solved for $x_{n-1}$. Working upward through the upper-triangular system in this manner, one can solve it for all the unknowns $x_n$ through $x_1$. This algorithm and its relatives have the generic name Gaussian elimination and back-solution. It is easy to code the algorithm directly in the language of your choice, but that straightforward exercise is not a desirable one, as we shall see immediately below.

The procedure just described has to be modified a bit under certain circumstances. For example, it can happen that after suitable multiples of the first equation have been subtracted from the second through $n$-th to zero out their $x_1$ terms, the (new) second equation has no $x_2$ term. Under the assumption that the matrix A of coefficients of the system is invertible, at least one of the new equations must have a nonzero $x_2$ term. (Otherwise the $(n - 1) \times (n - 1)$ submatrix in which one is working would have a first column consisting entirely of zeros, so there would be a nonzero $(n - 1)$-dimensional vector $(x_2, \ldots, x_n)^T$ which that submatrix annihilated. If we hold these $x_i$’s fixed we can obviously find a first coördinate...
for which the nonzero \( n \)-dimensional vector \( (x_1, x_2, \ldots, x_n)^T \) satisfies \( a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = 0 \) [the homogeneous equation associated with the first equation of the original system]; but then the vector \( (x_1, x_2, \ldots, x_n)^T \) satisfies the homogeneous equation \( Ax = 0 \) associated with the [whole] original system!\(^{[2]}\), contrary to the assumed existence of \( A^{-1} \).) The equation with the nonzero \( x_2 \) term can be interchanged with the second equation (an operation called \textit{pivoting}), after which the algorithm proceeds as described in the preceding paragraph. Such preliminary interchanges can be performed before any full “loop” of the algorithm, provided that the \( k \)-th equation is only interchanged with an equation “below” it, i.e., with the \((k+1)\)-st through \( n\)-th equation. Burden \& Faires give an example of the algorithm—in a case chosen so that pivoting becomes necessary at the beginning of the second loop—as Ex. 2, p. 356 ff. They also write out the algorithm—both in their pseudo-code and in Maple—on pp. 358-359, and they show you the operation counts, all of which are \( O(n^3) \), on pp. 360-362. Note that—from the standpoint of turning the pseudo-code into actual code—Step 3 of Algorithm 6.1 is clumsy, and Burden \& Faires do not include this step in their Algorithm 6.2. Instead, they (and everybody else) do the following. Assuming that the matrix of coefficients of the system was stored in an array \( A[I,J] \), one starts a run by initializing an array \( \text{NROW}[I] \) so that \( \text{NROW}[I] = I \), \( I = 1, \ldots, N \). Then one looks at the “virtual system” for which the coefficients of the \( I\)-th equation are \( A[NROW[I],J], \ I = 1, \ldots, N \) and the right-hand side is \( B[NROW[I]] \). If at some step of the algorithm one finds it desirable to interchange the \( I_1\)-st and \( I_2\)-nd equations, one simply swaps the contents of \( \text{NROW}[I_1] \) and \( \text{NROW}[I_2] \). The algorithm, which sees only the indices \( I \) and \( J \), is then looking at a system for which interchange is not necessary. A similar approach to column indices can be used to interchange “virtual unknowns” if one wants to do maximal (or complete) pivoting: see below, or see Burden \& Faires p. 374.

Atkinson discusses pivoting (and scaling) in his §8.2, p. 515 ff.: Burden \& Faires discuss various simple approaches to the question of whether pivoting is necessary or desirable in their §6.2. Methods for selecting equations to pivot are called \textit{pivoting strategies}. The simplest strategy, and one that usually works, is: prior to the \( I\)-th loop through the elimination process, interchange the (virtual) \( I\)-th equation with the first (virtual) equation among those below and including the \( I\)-th whose \( I\)-th-column coefficient is as large as possible in absolute value. This strategy is called the \textit{maximal column} pivoting strategy and is implemented in Burden \& Faires’ Algorithm 6.2. Another strategy begins by finding the “size” \( S[I] \), given by \( \max \{ |A[I,J]| : 1 \leq J \leq N \} \), of the \( I\)-th equation and then, at each step, interchanging the (virtual) \( I\)-th equation with the first (virtual) equation among those below and including the \( I\)-th whose \( I\)-th-column coefficient is \textbf{as large as possible in absolute value in comparison with its own} \( S[I] \). Since the Gaussian-elimination algorithm is really a recursion, in principle \( S[\cdot] \) would need to be updated at each trip around the loop: the lower \((n-k) \times (n-k)\) matrix will have changed considerably, and that will probably have changed the row in which the relatively largest pivot-candidate is to be found (in the first nonzero column). The partial pivoting strategy of Atkinson’s §8.2, \textbf{Definition 1}, p. 515 \textit{does} include this updating step, as does Van Loan’s discussion of and algorithm for pivoting in his §6.3.4, p. 227. There is some numerical-analysis folklore asserting that it suffices (in most cases) to compute the maxima \( S[\cdot] \) once-and-for-all; that is what is actually done under the name \textit{implicit scaling} in Atkinson’s §2, p. 519 and is also done in Burden \& Faires’ Algorithm 6.3, p. 371, demonstrated as their Ex. 4, same page. Thus Atkinson’s algorithm on p. 520 is not quite partial pivoting.

Note that it is also possible to interchange unknowns (again one does this virtually, so that the machine sees an array \( A[NROW[I],\text{NCOL}[J]] \)) and thereby to move the absolute-largest element in the (virtual) array \( \{A[NROW[I],\text{NCOL}[J]] : K \leq I \leq N, \ K \leq J \leq N \} \) to the pivot position. This approach, strategy, and algorithm are called \textit{complete pivoting}: when a distinction is necessary, the pivoting strategies we have considered above go by the name of \textit{partial pivoting}. Coding the complete-pivoting algorithm is a fairly straightforward exercise. Unfortunately, for each of these pivoting strategies it is possible to construct systems that the strategy cannot handle very well. A more complete discussion-with-references than we plan to give of the growth of error using various pivoting strategies can be found in Golub \& Van Loan, §3.4, p. 108 ff.

\begin{enumerate}
\item \textbf{2. A Note on Matrix Inversion.} One generally counsels students of numerical analysis \textit{not} to think much about computing the inverses of matrices (except, of course, when special problems make it necessary to do that). In almost all cases, the amount of computational effort required to solve \( Ax = b \)
— or even to solve $Ax = b_r$ for many different values of the r. h. s. $b_r$ is small in comparison with that of computing $A^{-1}$, provided that the computational work involved in Gaussian elimination is saved as a matrix factorization (see below) and that number of $\ell$’s is small in comparison with $n$ (as is the case in most “real-world” problems). For hand computation of $A^{-1}$, most students in Math 250 learn the algorithm that goes “form the partitioned matrix $[A : I]$ and do row operations on it until it has the form $[I : B]$—then $B$ will be $A^{-1}$.” That algorithm is basically a modification of the Gauss-Jordan method described at the beginning of Atkinson’s §8.3, pp. 522–523; this method is also given as Burden & Faires’ Exercise 8, p. 364 and the algorithm given as Ex. 5, pp. 383–384; it is a perfectly adequate algorithm for hand computation of inverses. From the point of view of machine computation, however, one wants to avoid the computation of explicit inverses. See the discussion of “the LU mentality” that forms §6.3.5 of Van Loan (pp. 230–231): always think “solve the system” instead of “invert the matrix,” even when you have to think “invert the matrix” to see where the system comes from.

3. Gaussian Elimination is Matrix Factorization. This fact is important for theoretical purposes, but it also enables one to solve families of equations $Ax = b$ for “many choices of $b$ but always the same coefficients $A$” in an extremely efficient way. The basic idea is easy to state if one can wave one’s hands a lot; saying it in notation is harder. Read Atkinson’s §8.1 (p. 508 ff.), Burden & Faires’ §6.5 (p. 394 ff.), Kincaid & Cheney’s §4.1 ff. (beginning on p. 148), Van Loan’s Ch. 6 or this treatment; you’ll develop your own understanding of the process.

It helps to know that every elementary row operation that can be done on $n \times n$ matrices can be done by a left multiplication. Recall that the elementary row operations are of three types: (1) multiplying the $i$-th row by a constant $\lambda$; (2) multiplying the $i$-th row by a constant $\lambda$ and adding the product to the $k$-th row; (3) interchange of rows. The proof that these operations can be accomplished on a matrix $A$ by multiplying it by an appropriate $E$ consists in observing that if it is true, then one can figure out what $E$ has to be by applying it to the partitioned matrix $[A : I]$ and seeing what happens on the r. h. s. of the partition. For example, multiplying the $i$-th row of the partitioned matrix by $\lambda$ results in the matrix

\[
\begin{pmatrix}
1 & \cdots & 0 & \cdots & 0 \\
0 & \ddots & \vdots & \ddots & \vdots \\
i & 0 & \lambda & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \cdots & 1
\end{pmatrix}
\]

and multiplying by this matrix—on the left—will obviously multiply the $i$-th row of any given $(n \times n)$ matrix $A$ by $\lambda$. Similarly, multiplying $A$ on the left by the matrix

\[
E_{i,k}(\lambda) = \begin{pmatrix}
1 & \cdots & 0 & \cdots & \cdots & 0 \\
0 & \ddots & \vdots & \ddots & \ddots & \vdots \\
i & 0 & \cdots & 1 & \cdots & \cdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & \lambda & \cdots & 1 & \cdots \\
0 & \cdots & \cdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \cdots & \cdots & 1
\end{pmatrix}
\]

will multiply the $i$-th row of $A$ by $\lambda$ and add it to the $k$-th row. Note that since it is obvious that the inverse of that row operation is the operation of multiplying the $i$-th row of $A$ by $-\lambda$ and adding that to the
the $k$-th row, the inverse of the matrix $E_{i,k}(\lambda)$ that we just considered can be produced by a simple sign change:

$$E_{i,k}(\lambda)^{-1} = \begin{pmatrix} 1 & \cdots & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \vdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & 1 & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & -\lambda & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \ddots & 1 \end{pmatrix} = E_{i,k}(-\lambda).$$

That observation is crucial to understanding how Gaussian elimination is the same thing as a certain kind of factorization of the matrix of coefficients, so we shall postpone consideration of elementary row operations of type (3) above (row interchanges) for a moment and get right to work on Gaussian elimination (these will not employ pivoting, since it is that for which one needs row interchanges).

Consider the operations in Gaussian elimination that pertain to the first column of $A$. Let $A_i$ denote the $i$-th row of $A$. One finds the multipliers $m_{21}, m_{31}, \ldots, m_{n1}$ for which the coefficient of $x_1$ in the row (or equation) $A_i - m_{11}A_1$ is zero, and subtracts (the fact that one thinks of the multipliers as related to \textit{subtraction} is crucial, because one wants to think about multiplying by the \textit{inverses} of matrices). Indeed, from the standpoint of the matrices $E_{i,k}(\cdot)$ we found above, the matrix of coefficients after the first loop of Gaussian elimination—the one in which $a_{11}$ will be the only nonzero coefficient in the first column—is given by

$$E_{1,n}(-m_{11}) \cdots E_{1,2}(-m_{21})A = E_{1,n}(m_{11})^{-1} \cdots E_{1,2}(m_{21})^{-1}A = \left[E_{1,2}(m_{21}) \cdots E_{1,n}(m_{n1})\right]^{-1}A.$$

Let us denote the matrix product $[E_{1,2}(m_{21}) \cdots E_{1,n}(m_{n1})]$ by $E_1$ — it (or more accurately its inverse) does all the Gaussian elimination that involves column 1 of $A$. We can similarly denote the product $[E_{2,2}(m_{22}) \cdots E_{2,n}(m_{n2})]$ by $E_2$, and so on all the way down to $E_{n-1} = E_{n-1,n}(m_{n,n-1})$. With this abbreviated notation, the upper-triangular matrix $U$ that results at the conclusion of Gaussian elimination has the form

$$U = E_{n-1}^{-1} \cdots E_1^{-1}A.$$

If we peel off the matrices on the r. h. s. of the equation just given—multiply the equation first by $E_{n-1}$ (on both sides, of course), then by $E_{n-2}$, etc.—we see that it is equivalent to the equation

$$A = \left[ E_1E_2 \cdots E_{n-2}E_{n-1}I \right] U,$$

where we have put in the brackets and the “unnecessary” identity matrix for reasons which (we hope) will soon become clear. Consider the row operation that left multiplication by $E_{n-1} = E_{n-1,n}(m_{n,n-1})$ performs on $I$. It multiplies the $(n-1)$-st row of $I$ — whose only nonzero entry is the “1” on the diagonal — by $m_{n,n-1}$ and adds it to the $n$-th = bottom row. So the product continues to have only “1”’s on the diagonal, has no nonzero entries above the diagonal, and has $m_{n,n-1}$ in the $(n,n-1)$ position. The next factor, $E_{n-2}$, adds the $m_{i,n-2}$ multiplies of the $(n-2)$-nd row of the product so far formed to the $i$-th row for $n-1 \leq i \leq n$: but since that $(n-2)$-nd row was all zeros except for the “1” on the diagonal, the effect is again to put the multipliers $m_{i,n-2}$ into exactly the positions that their indices label, \textit{while not touching elements of any other column in the product matrix}. The products continue to build up in this way until the last multiplication

$$E_1 \cdot \left[ E_2 \cdots E_{n-2}E_{n-1}I \right]$$
puts the column of multipliers \((m_{21}, m_{31}, \ldots, m_{n1})^T\) into the first column of the product. Since all the row operations have been done strictly below the diagonal of the “unnecessary” factor \(I\), the product is a lower-triangular matrix. It follows that if we follow the usual notational convention and call this product \(L\), then we shall finally have

\[
L = E_1E_2E_3 \cdots E_{n-2}E_{n-1} = \begin{pmatrix}
1 & 0 & \cdots & 0 \\
m_{21} & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
m_{n1} & m_{n2} & \cdots & 1
\end{pmatrix}
\]

and so

\[
A = \left[ E_1E_2 \cdots E_{n-2}E_{n-1} \right] U = LU.
\]

What we have discussed is exactly the material of Atkinson’s §8.1 (except for operation counts), Burden & Faires’ §6.5 (we have given the proof of their Theorem 6.18 [pp. 394 ff.]), or Van Loan’s §6.3. Somewhat more exploded views of the matrices involved may be found on pp. 395–396 of Burden & Faires. Their Example 1 on p. 397, like Atkinson’s treatment of his system (8.1.4) on p. 510 ff., lets the reader actually see the triangular factorization coming into being as the steps of “elementary” Gaussian elimination are carried out in a concrete numerical case.

We postponed consideration of row interchanges—pivoting—above, but we can return to it now. It is routine to verify that if \(\pi\) is a permutation of the indices \(\{1, 2, \ldots, n\}\) and it is applied to the indices of the rows of the identity matrix \(I = [\delta_{ij}]\), the result is the matrix \(P [\delta_{\pi(i),j}]\), and that then if \(A\) is any \(n \times n\) matrix, then \(PA = [a_{\pi(i),j}]\).

To see what is going on in Gaussian elimination when pivoting takes place, it is convenient to look at Gaussian elimination with pivoting from the viewpoint of a conceptual computing machine like the one we imagined on pp. 1–2 of these notes, with the matrix \(A\) stored in an array \(A[I, J]\) (cf. Burden & Faires’ Algorithm 6.2). Regardless of what pivoting strategy is employed, when Gaussian elimination is completed the machine is looking at an \(LU\)-factorization of the matrix \(PA = [a_{\text{NROW}(i),j}]\).

So we can at least say that given any invertible square matrix \(A\) there will exist an \(LU\)-factorization for some version of \(A\) whose rows have been permuted. (Indeed, it will soon be clear that some invertible matrices cannot have \(LU\)-factorizations: we shall show that if it is possible to write \(A = LU\), then Gaussian elimination can be performed on \(A\) without pivoting, and that is manifestly impossible for such simple examples as

\[
\begin{pmatrix}
0 & -1 \\
1 & 0
\end{pmatrix}.
\]

For a discussion of getting all the pieces of \(PA = LU\) directly from the built-in Matlab function \texttt{LU} (and of what goes on inside the Matlab operator \("\backslash"\) when one computes the solution of \(Ax = b\) as \(x = A\backslash b\)), see Van Loan’s §6.3.6, p. 231 ff.

4. Additional Consequences of Gaussian Elimination and of Factorization. The following material is mostly of theoretical importance, but knowing it can help one see what one is doing.

**Proposition:** Let \(A = [a_{ij}]\) be an \(n \times n\) matrix. The following conditions are logically equivalent:

1. Gaussian elimination, or factorization, can be performed on \(A\) without the necessity of row interchanges.
2. Gaussian elimination, or factorization, can be performed on every leading principal diagonal submatrix \(A_k = \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{pmatrix}\) of \(A\), without the necessity of row interchanges.
3. Every \(A_k\) is nonsingular.
(4) A factorization \( A = LU \) holds for some lower-triangular \( L \) and some upper-triangular \( U \).

(5) For each \( 1 \leq k \leq n \), a factorization \( A_k = \hat{L}_k \hat{U}_k \) holds for some lower-triangular \( \hat{L}_k \) and some upper-triangular \( \hat{U}_k \).

Proof. (1) \( \Rightarrow \) (2) is quite clear: the requirements of (2) include (1), and on the other hand if Gaussian elimination can be performed on \( A \) without row interchanges, then the very same row operations that do elimination on \( A \) will simultaneously perform Gaussian elimination on the submatrices \( A_k \) for \( 1 \leq k \leq n \). This also shows that (1) \( \Rightarrow \) (5).

(2) \( \Rightarrow \) (3) is clear for a number of reasons; perhaps the quickest is that the \( A = LU \) factorization also gives \( A_k = L_k U_k \), and since the upper- and lower-triangular factors are nonsingular so is \( A_k \). We have actually looked briefly (on p. 1) above at the reason that (3) \( \Rightarrow \) (2): if (2) were false and pivoting were to be required because the entry in position \((k,k)\) equaled 0 after the \((k-1)\)-st trip through the Gaussian elimination loop, then the current state of the “working matrix” of \( A \) would have the appearance

\[
\begin{bmatrix}
  u_{11} & \cdots & u_{1k} & \vdots & u_{1,k+1} & \cdots & u_{1n} \\
  0 & \ddots & \vdots & \ddots & \ddots & \ddots & \ddots \\
  0 & \cdots & 0 & \ddots & u_{k,k+1} & \cdots & u_{kn} \\
  0 & \cdots & u_{k+1,k} & \cdots & u_{k+1,k+1} & \cdots & u_{k+1,n} \\
  \vdots & \cdots & \vdots & \cdots & \cdots & \cdots & \vdots \\
  0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots 
\end{bmatrix}
\]

The \( k \times k \) leading principal submatrix of this matrix has a row of zeros and is thus certainly singular; since it was obtained from \( A_k \) by elementary (and thus invertible) row operations, \( A_k \) must also have been singular, so (3) cannot hold.

We have already seen that (1) \( \Rightarrow \) (5), and of course (5) \( \Rightarrow \) (4). Since (4) implies that the leading principal submatrices of \( A \) are nonsingular (the triangular matrices \( \hat{L}_k \) and \( \hat{U}_k \) being necessarily nonsingular), we have (4) \( \Rightarrow \) (3) and that completes the circle of implications.

As Atkinson observes in his Corollary on p. 514, Gaussian elimination furnishes a fairly efficient way to compute the determinant of a square matrix. If a matrix has the form \( LU \) where \( L \) is lower-triangular with 1’s on its diagonal, then successive expansion by minors of the top row on smaller and smaller matrices shows that \( \det L = 1 \). Similarly, when \( U \) is upper-triangular with diagonal entries \( u_{11}, u_{22}, \ldots, u_{nn} \), successive expansion by minors of the left column on smaller and smaller matrices shows that \( \det U = \prod_{i=1}^{n} u_{ii} \). (Cf. Burden & Faires, Theorem 6.16, p. 390.) If Gaussian elimination was performed on \( A \) without pivoting to yield the factorization \( A = LU \), then, we have \( \det A = \prod_{i=1}^{n} u_{ii} \) and the factors of the product on the r. h. s. are stored in easily accessible places. If pivoting was required, though, the factorization actually has the form \( PA = LU \) and the sign of the product may be wrong. If one has done Gaussian elimination/LU-factorization using virtual row interchanges, however, it is easy to compute the sign change. The permutation of the rows is built up one row-interchange at a time in the algorithm. Since interchanging rows of a matrix reverses the sign of its determinant, each virtual swap of rows reverses the sign of the determinant of the virtual matrix. If we initialize a real location \( \text{SIGN} \) with 1 when the program begins, and execute \( \text{SIGN} = -\text{SIGN} \) every time our pivoting strategy forces a row swap, then at the end of execution \( \text{SIGN} \) will contain \((-1)^q\), where \( q \) is the number of row interchanges whose composition is \( P \). As a result we shall have \( \det P = \text{SIGN} \), so

\[
\text{SIGN} \det A = \det PA = \det LU = \prod_{i=1}^{n} u_{ii} = \det A
\]
since \( \text{SIGN} \) is its own multiplicative inverse. So keeping track of the sign is no problem. If one wants to do complete pivoting and also interchange columns of the virtual matrix, the same approach works. (To see that the considerations of this paragraph can be helpful, consider the count of the number of operations required to evaluate a determinant from the definition or by minors! These are floating-point operations and therefore inexact in general, so keeping operation counts down is important for the sake of accuracy, regardless of the speed of computation.)

There is a standard trick for minimizing storage requirements in Gaussian elimination built in to Atkinson’s version of the algorithm on pp. 520–521; it may be worth while to look at it explicitly. Suppose one is doing Gaussian elimination—but for now assume that one is doing it without pivoting, so \( \text{NROW}(i) = i \) for all \( 1 \leq i \leq n \). Suppose one is currently pivoting on the \( i \)-th row = in the \( i \)-th column. Once a multiplier \( m_{ji} \) had been found for some \( j > i \) and the subtractions down the \( i \)-th column carried out, the location \( a(j, i) \) would contain a zero forever after. That location is therefore available to store the multiplier \( m_{ji} \), so one stores it there. When the algorithm terminates, the locations in a \( U \) matrix: if \( \text{operation of multiplying the upper- or lower-triangular matrix whose diagonal entries equal 1} \). This follows from the fact that the row or lower-triangular matrix can be written as a product of an invertible diagonal matrix with an invertible matrix. It is invertible, so all of its diagonal entries are nonzero, and its inverse is the diagonal matrix whose entries are \( 1 \)'s. The assertions about upper-triangular matrices now follow by taking transposes.

It is easy to verify (in all cases) that if \( \text{the column index} \) hold \( m_{ij} \) entries of \( U \) diagonal element in the \( ii \)-position of the product \( LM \) is \( 1 \), the diagonal entries of \( L \) are 1's, then since \( \text{D} \) is a diagonal matrix. It is invertible, so all of its diagonal entries are nonzero, and its inverse is the diagonal matrix whose \( ii \)-position entry is \( 1/\text{di} \). It follows in all cases that \( \text{D}^{-1}\text{L}_1\text{L} = \text{I} \), so \( \text{D}^{-1}\text{L}_1 = \text{L}^{-1} \) is lower-triangular. It is easy to verify (in all cases) that if \( \text{L} = \begin{bmatrix} \ell_{ij} \end{bmatrix} \) and \( \text{M} = \begin{bmatrix} m_{ij} \end{bmatrix} \) are two lower-triangular matrices, then the diagonal element in the \( ii \)-position of the product \( \text{LM} \) is \( \ell_{ii}m_{ii} \). It follows that if the diagonal entries of \( \text{L} \) are 1’s, then since \( \text{L}^{-1}\text{L} = \text{I} \) and the diagonal entries of \( \text{I} \) are 1’s, the diagonal entries of \( \text{L}^{-1} \) must also be 1’s. The assertions about upper-triangular matrices now follow by taking transposes.

It probably isn’t worth setting this fact off as a formal proposition, but it is a fact: any invertible upper- or lower-triangular matrix can be written as a product of an invertible diagonal matrix with an invertible upper- or lower-triangular matrix whose diagonal entries equal 1. This follows from the fact that the row operation of multiplying the \( i \)-th row of a matrix can be implemented by left multiplication by a diagonal matrix: if \( \text{U} = \begin{bmatrix} u_{ij} \end{bmatrix} \) is upper-triangular, say, then

\[
U = \begin{bmatrix}
  u_{11} & 0 & \cdots & 0 \\
  0 & u_{22} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & u_{nn}
\end{bmatrix} = \begin{bmatrix}
  1 & u_{12}/u_{11} & \cdots & u_{1n}/u_{11} \\
  0 & 1 & \cdots & u_{2n}/u_{22} \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & 1
\end{bmatrix}.
\]
The same thing can be done with columns, so that one can write \( U = U_0 D \) just as well as \( U = D U_1 \). And of course the same thing can be done for lower-triangular matrices, but the ones we work with usually come to us with 1’s on their diagonals.

**Proposition:** If an (invertible) matrix \( A \) can be factorized in the form \( A = LDV \), where the first factor is lower-triangular with 1’s on its diagonal, the second factor is diagonal (with nonzero diagonal entries), and the third factor is upper-triangular with 1’s on its diagonal, then the factors in the factorization are uniquely determined by \( A \).

**Proof.** Assume two factorizations \( LDV = L_1 D_1 V_1 \). Then \( L_1^{-1} LD = D_1 V_1 V^{-1} \). Since the matrix on the l. h. s. is lower- and the matrix on the r. h. s. upper-triangular, their common value is a diagonal matrix. Moreover, since \( L_1^{-1} L \) has 1’s on its diagonal and \( V_1 V^{-1} \) has 1’s on its diagonal, their common value is equal both to \( D \) and to \( D_1 \), which are therefore equal. Multiplying both sides on the right by \( D^{-1} \) gives \( L_1^{-1} L = D V_1 V^{-1} D^{-1} \) and since the matrix on the l. h. s. is lower-triangular with 1’s on the diagonal and the matrix on the r. h. s. is upper-triangular, the matrix on the l. h. s. is the identity and thus \( L = L_1 \). Multiplying \( I = DV_1 V^{-1} D^{-1} \) by \( D^{-1} \) on the left and \( D \) on the right similarly leads to \( V = V_1 \).

It follows from this proposition that, loosely speaking, the only ambiguity in \( LU \)-factorization comes from moving things back and forth between the diagonals of the lower- and the upper-triangular factors. In particular, the factors in an \( A = LU \) factorization in which \( L \) is required to have 1’s on its diagonal are uniquely determined by \( A \). Thus no matter how such a factorization is obtained, it will be identical to the one obtained by the Gaussian-elimination algorithm.

If we are to proceed into consideration of the special types of matrices, it would be helpful to know something about where these special classes come from. To do that efficiently, we may as well begin to look at the material of Chapter 7; it explains where some of these classes come from, and helps one to understand how the errors stemming from roundoff are controlled in computational linear algebra.


**Definition:** The \( \ell^p \) norms of vectors \( x = (x_1, x_2, \ldots, x_n)^T \) in \( \mathbb{R}^n \) or \( \mathbb{C}^n \) are defined by

\[
\| (x_1, \ldots, x_n)^T \|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}
\]

for \( 1 \leq p < \infty \). For \( p = \infty \), the \( \ell^\infty \) norm of a vector \( x = (x_1, \ldots, x_n)^T \) is defined by

\[
\| (x_1, \ldots, x_n)^T \|_\infty = \max \{ |x_i| : 1 \leq i \leq n \}.
\]

The subscript “\( \infty \)” is used because of the fact—not difficult to prove—that \( \lim_{p \to \infty} \| (x_1, \ldots, x_n)^T \|_p = \| (x_1, \ldots, x_n)^T \|_\infty \). (To prove it, factor the largest \( |x_k| \) out of the defining expression for \( \| x \|_p \) and watch what happens as \( p \to \infty \), paying particular attention to the case in which more than one \( |x_i| \) takes the maximum value.)

Atkinson shows you some of the geometry of these distance-measuring devices in Figure 7.2, p. 482 (see also Kincaid & Cheney, p. 200). Consider the \( \ell^1 \) unit ball for a moment: in two dimensions, it is the symmetric, diamond-shaped object (in fact, a square of sides whose Euclidean length is \( \sqrt{2} \)) obtained by reflecting the region bounded by the nonnegative \( x \) - and \( y \)-axes and the line \( x + y = 1 \) over both axes and the origin. In three dimensions, it is the octahedron obtained by reflecting the region bounded by the nonnegative \( x \)-, \( y \)- and \( z \)-axes and the plane \( x + y + z = 1 \) over all coordinate planes and the origin, so the fact that the \( \ell^1 \) and \( \ell^\infty \) unit balls in dimension 2 are similar is misleading; in \( n \) dimensions the sides of the unit ball in \( \ell^\infty \) correspond to the corners of the unit ball in \( \ell^1 \) rather than to its sides. The number \( p \) is called the exponent of an \( \ell^p \) norm; two numbers \( p \) and \( q \) for which \( (1/p) + (1/q) = 1 \) are called conjugate exponents (of each
other), and by special dispensation 1 and $\infty$ are said to be conjugate also. It is occasionally useful to know that

the following Hölder inequality is true: if $p$ and $q$ are conjugate exponents and $x = (x_1, \ldots, x_n)^T$ and $y = (y_1, \ldots, y_n)^T$, then the inequality

$$\left| \sum_{i=1}^n x_i y_i \right| \leq \sum_{i=1}^n |x_i y_i| \leq \|x\|_p \|y\|_q$$

holds. (There is a standard convex-geometry proof of this inequality, and it is given in almost everybody’s beginning real-analysis course: see, e.g., R. L. Wheeden and A. Zygmund, Measure and Integral, Marcel Dekker (1977), pp. 127–128, and substitute finite summation for integration.) This inequality is sharp in the sense that given any $x$ one can find a $y$ with $\|y\|_q = 1$ for which the inequality becomes an equality. In the case $p = q = 2$ one can simply take $y = x/\|x\|_2$, the “unit vector pointing in the direction of $x$”. In the case $p = 1$, $q = \infty$ one takes $y = (\pm 1, \ldots, \pm 1)^T$, where the sign of the $i$-th coordinate is chosen to be the same as that of $x_i$. In the case $p = \infty$, $q = 1$ one chooses an $i$ for which $|x_i| = \max \{|x_i| : 1 \leq i \leq n\}$; then one takes $y = (0, \ldots, \pm 1, \ldots, 0)^T$, where the single nonzero coordinate is the $i$-th and the sign is chosen to give a positive “dot product”.

The only values of $p$ (or $q$) that will turn up in this course are 1, 2, and $\infty$. The case $p = q = 2$ is the usual Euclidean norm (or distance) with which everyone is familiar from sophomore calculus. It is intimately connected with the dot product or inner product of vectors, about which we shall have more to say later. In the context of numerical analysis, norms are used (for the most part) in order to talk about (1) giving a positive “dot product” connected with the usual Euclidean norm (or distance) with which everyone is familiar from sophomore calculus. It is intimately connected with the dot product or inner product of vectors, about which we shall have more to say later. In the context of numerical analysis, norms are used (for the most part) in order to talk about (1) giving bounds for error and (2) convergence of sequences generated by algorithms. As Atkinson shows in Thm. 7.7, p. 483, all norms on $\mathbb{R}^n$ or $\mathbb{C}^n$ are equivalent with respect to convergence. Since we are only going to work with $p$-norms for $p = 1, 2$ and $\infty$, we can sharply exhaust the following inequality, in which sharp explicit numerical values are given for the “$c_1$” and “$c_2$” of Atkinson’s 7.7:

**Proposition:** For any vector $x$ in $\mathbb{R}^n$ or $\mathbb{C}^n$,

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq n \cdot \|x\|_\infty .$$

**Proof.** The fact that $\|x\|_\infty \leq \|x\|_2$ is obvious (why?). If $e_j$ denotes the $j$-th of the usual “standard basis vectors” $e_j = (0, \ldots, 1, \ldots, 0)^T$, then for $x = (x_1, x_2, \ldots, x_n)^T$ one can write for any $p$ with $1 \leq p \leq \infty$

$$\|x\|_p = \left( \sum_{j=1}^n |x_j e_j|_p \right) \leq \sum_{j=1}^n |x_j| \|e_j\|_p$$

$$= \sum_{j=1}^n |x_j| \cdot 1 = \|x\|_1$$

where the inequality at the end of the first line follows from the triangle inequality and the “absolute homogeneity” of all norms—property N2 on p. 481—and the fact that $\|e_j\|_p = 1$ for every $p$. Finally, since $|x_j| \leq \|x\|_\infty$ holds for every index $1 \leq j \leq n$, we have for every $1 \leq p < \infty$

$$\|x\|_p^p = \sum_{j=1}^n |x_j|^p \leq \sum_{j=1}^n \|x\|_\infty^p = n \cdot \|x\|_\infty^p$$

$$\|x\|_p \leq \sqrt[n]{n} \cdot \|x\|_\infty$$

of which $\|x\|_1 \leq n \|x\|_\infty$ is a particular case. To see that this is sharp, consider the vector $(1, \ldots, 1)$.

The natural, induced or operator norms on the spaces of $n \times n$ matrices (also called matrix norms by some authors, e.g., Kincaid & Cheney) are discussed adequately by Atkinson on pp. 484–486, though he does not mention two important and frequently used facts about operator norms. The first is that the operator norm of an identity matrix is always 1, because

$$\|I\| = \max \{\|Ix\| : \|x\| \leq 1\} = 1$$

9
is trivially true. The second is that the natural norm is the “amplification factor” for \( A \), in that \( \|A\| \) is the smallest number \( M \geq 0 \) for which the inequality \( |Ax| \leq M \cdot \|x\| \) will hold for every \( x \) in \( \mathbb{R}^n \) or \( \mathbb{C}^n \). Reason: on one hand, any such number \( M \) is larger than \( \|A\| \), because

\[
\|Ax\| \leq M \cdot \|x\| \quad \Rightarrow \quad \max \{ \|Ax\| : \|x\| \leq 1 \} \leq M \cdot 1 = M ;
\]

on the other hand, \( \|A\| \) is such a number, because for any \( 0 \neq x \) in \( \mathbb{R}^n \) or \( \mathbb{C}^n \) we may set \( z = (1/\|x\|) \cdot x \); then \( \|z\| = 1 \) and

\[
\|(1/\|x\|)Ax\| = \|Az\| \leq \|A\| \quad \text{by definition}
\]

\[
\|Ax\| \leq \|A\| \cdot \|x\| \quad \text{upon clearing fractions}.
\]

Atkinson’s unworked example (7.3.17) on p. 487, giving a concrete computable form for \( \|A\|_\infty \), is one of the happiest facts in numerical analysis. The concrete expression (7.3.19) for the natural matrix norm associated with the \( \ell^2 \) norm of vectors is also useful for thinking, but its numerical value for a particular matrix is much harder to compute. (For further information on these topics see also Kincaid & Cheney p. 201 ff., and their §5.4 or the discussion of singular value decompositions below.)


The concrete expression

\[
\|A\|_\infty = \max \{ \sum_{j=1}^{n} |a_{ij}| : 1 \leq i \leq n \}
\]

for the natural matrix norm associated with the \( \ell^\infty \) norm of vectors is the tip of an iceberg of information; much of it is denoted by names involving the Russian mathematician S. A. Geršgorin, who seems to have been among the first to exploit this computation. A convenient formulation of the most basic fact is

**Geršgorin’s Theorem** (one version): Let \( A = [a_{ij}] \) be an \( n \times n \) matrix of real or complex numbers. The **(strict) diagonal dominance** condition (cf. Atkinson, formula (8.6.8), p. 546; Burden & Faires, Def. 6.19 and Theorem 6.20, pp. 404–405; Kincaid & Cheney, p. 188 ff.)

\[
|a_{ii}| > \sum_{j \neq i} |a_{ij}|
\]

is sufficient for the invertibility of \( A \). Moreover,

\[
\|A^{-1}\|_\infty \leq \frac{1}{\min \{ |a_{ii}|-\sum_{j \neq i} |a_{ij}| : 1 \leq i \leq n \}}.
\]

**Proof.** Suppose the relation \( Ax = y \) holds, where \( x = (x_1, \ldots, x_n)^T \) and \( y = (y_1, \ldots, y_n)^T \). Let \( \ell \) be an index for which \( |x_\ell| = \max \{ |x_i| : 1 \leq i \leq n \} = \|x\|_\infty \). Then we can write (using the triangle inequality for real or complex numbers in the form \( |a| = |a-b| + |b| \), which implies \( |a| - |b| \leq |a-b| \)):

\[
\|y\|_\infty \geq |y_\ell| = \sum_{j=1}^{n} a_{\ell j} x_j = |a_{\ell \ell} x_\ell - \sum_{j \neq \ell} (-a_{\ell j}) x_j| \geq |a_{\ell \ell} x_\ell| - \sum_{j \neq \ell} |a_{\ell j}| |x_j| = |a_{\ell \ell}| - \sum_{j \neq \ell} |a_{\ell j}| \cdot |x_\ell| = \|A\|_\infty \cdot |x_\ell|.
\]

The first thing this inequality tells us is that if \( y = 0 \), then—since \( |a_{\ell \ell}| - \sum_{j \neq \ell} |a_{\ell j}| > 0 \)—one must have \( \|x\|_\infty = 0 \) and therefore \( x = 0 \). In other words, \( Ax = 0 \) implies \( x = 0 \): \( A \) cannot be singular, and hence it is invertible. Next, since we now know that \( A^{-1} \) exists, we can let \( x = A^{-1} y \) in the inequality and get

\[
\|y\|_\infty \geq |a_{\ell \ell}| - \sum_{j \neq \ell} |a_{\ell j}| \cdot \|A^{-1} y\|_\infty.
\]
The choice of \( \ell \) depends on \( y \), but if we use the \( \ell \) for which the difference \( |a_{\ell \ell}| - \sum_{j \neq \ell} |a_{\ell j}| \) is minimum (but still positive, by hypothesis), we can get a constant \( m = \min \{ |a_{ii}| - \sum_{j \neq i} |a_{ij}| : 1 \leq i \leq n \} \) for which \( \|y\|_{\infty} \geq m \cdot \|A^{-1}y\|_{\infty} \) holds for all \( y \). Then \( \|A^{-1}y\|_{\infty} \leq (1/m) \cdot \|y\|_{\infty} \) holds for all \( y \), so (specializing to the vectors \( y \) of norm \( \leq 1 \)) we see that \( \|A^{-1}\|_{\infty} \leq 1/m \), as the theorem asserted.

Remark: Under certain conditions, the conclusion of the Geršgorin theorem holds when the strict inequality \( > \) in the hypothesis is replaced by the weak inequality \( \geq \). We shall jump off this bridge when we come to it.

Presumably everyone knows what an eigenvalue is, so we may safely assert

**Corollary (Geršgorin Discs):** Every eigenvalue \( \lambda \) of a given \( n \times n \) matrix \( A = [a_{ij}] \) must lie in the union of the discs \( \{ \lambda : |\lambda - a_{ii}| \leq r_i \} \) in the complex plane centered on the diagonal elements of \( A \), where the radii \( r_i \) of the discs are given by

\[
r_i = \sum_{j \neq i} |a_{ij}|, \quad 1 \leq i \leq n.
\]

**Proof.** Suppose \( \lambda \) does not belong to any of those discs, and consider the matrix \( (\lambda I - A) \). Its diagonal elements are \( \lambda - a_{ii} \), and its off-diagonal elements are simply the \( -a_{ij} \). Since \( \lambda \) does not belong to any of those discs, we have

\[
|\lambda - a_{ii}| > \sum_{j \neq i} |a_{ij}| = \sum_{j \neq i} | -a_{ij} |
\]

for all \( 1 \leq i \leq n \); therefore \( (\lambda I - A)^{-1} \) exists by Geršgorin’s theorem, and so \( \lambda \) cannot be an eigenvalue of \( A \).

Needless to say, if one has to go on eigenvalue hunts it helps to have the Geršgorin discs to use in narrowing the field somewhat. Also, if \( \lambda \in \mathbb{C} \) lies outside all the Geršgorin discs, we get an estimate for \( \|A^{-1}\|_{\infty} \) by applying the Geršgorin theorem on the preceding page to \( (\lambda I - A) \).

Geršgorin’s theorem gives us a somewhat easier proof(1) of Burden & Faires’ Theorem 6.20, p. 404, which is namely: if \( A \) is a matrix satisfying the strict diagonal dominance condition, then all the new matrices produced in the course of applying Gaussian elimination to \( A \) are also strictly diagonally dominant, so that in particular they never have zero elements on their diagonals. We know that Gaussian elimination can be performed without row interchanges if (and only if) all the leading principal submatrices of \( A \) are nonsingular; since those submatrices inherit strict diagonal dominance from \( A \), they are indeed nonsingular and hence Gaussian elimination can be performed without row interchanges.

Before we can talk intelligently about symmetric real matrices (and their equally important complex analogues) we need to think a bit about the rôle of the dot product, or inner product, in studying matrices and vectors.

**7. Matrices and the Dot or Inner Product.** Given two vectors \( x = (x_1, \ldots, x_n)^T \) and \( y = (y_1, \ldots, y_n)^T \) in \( \mathbb{R}^n \), their **dot product** or **inner product** is defined(2) by the equation \( \langle x, y \rangle = \sum_{j=1}^{n} x_j y_j \). It is useful to realize that this expression is actually a matrix product, provided that \( 1 \times 1 \) matrices are interpreted as simply being numbers: we have

\[
\langle x, y \rangle = \sum_{j=1}^{n} x_j y_j = y^T x
\]

---

(1) Notes that prove this theorem (in a somewhat different way) will be posted when needed.

(2) Atkinson uses ordinary parentheses to denote the inner product—see the top of p. 468—where I am using angle brackets. Since the ordinary curved parentheses are used in inconsistent ways in so many contexts, it seems reasonable to give them a little less work to do. Cf. the many uses of the word “normal” that confuse so many discussions.
by the definition of matrix multiplication. This equation, and the fact that transposition of matrices reverses
the order of multiplication (see Atkinson p. 467 ff., Burden & Faires p. 383 ff., Kincaid & Cheney p. 1148
ff. or your favorite linear-algebra textbook for a review), enable us to see that transposition of matrices must
have some kind of geometric meaning: for any \( n \times n \) matrix \( A \), we have

\[
\langle Ax, y \rangle = y^T (Ax) = y^T Ax = y^T ((A^T)^T) x = (A^T y)^T x = \langle x, A^T y \rangle ;
\]

“A can pass across the inner product, but it gets transposed.” **Symmetric matrices**—those satisfying the
condition \( A = A^T \)—thus have the special property with respect to the inner product that they “look the
same on one side of the inner product as they do on the other.”

For the purpose of talking about eigenvalues it is convenient to think about vectors and matrices with
complex numbers as entries. The reason is that the eigenvalues of a matrix \( A \) are exactly the solutions of the
characteristic equation \( \det (\lambda I - A) = 0 \), which may have no real roots. A simple example is the
“90° rotation” matrix

\[
\begin{bmatrix}
0 & -1 \\
1 & 0
\end{bmatrix},
\]

whose characteristic equation \( \lambda^2 + 1 = 0 \) has the complex roots \( i \) and \( -i \)
but has no real roots. It can be shown that a polynomial equation with complex coefficients (and thus, in
particular, with real coefficients) always has as many complex roots, counted according to multiplicity, as
its degree allows. However, the example we just gave shows that finding the eigenvalues (and therefore also
presumably the eigenvectors) of a matrix with real entries may require the use of complex numbers.

Making a workable inner product for vectors with complex entries requires modifications that at first
may seem rather strange. We cannot use \( \langle x, y \rangle = \sum_{j=1}^{n} x_j y_j \) uncritically, because then—for example—the
nonzero vector \((1, i)^T \in \mathbb{C}^2 \) would be perpendicular to itself: \((1, i)(1, i)^T = 1^2 + i^2 = 1 - 1 = 0\), while we
would like the inner product of a vector with itself to be the square of its norm. So we introduce a “twist” in
the second factor\(^{(3)}\) and define for \( z = (z_1, \ldots, z_n)^T \) and \( w = (w_1, \ldots, w_n)^T \)

\[
\langle z, w \rangle = \sum_{j=1}^{n} z_j \overline{w}_j = w^* z
\]

where the **star**, adjoint or **Hermitean transpose** of a matrix or vector is defined to be its **conjugate transpose**:

\[
[a_{ij}]^* = [\overline{a}_{ji}].
\]

This inner product has all the properties of the familiar real inner product, including \( \|z\|^2 = \langle z, z \rangle \) and
the Cauchy-Schwarz inequality \( |\langle z, w \rangle| \leq \|z\| \|w\| \) (cf. Atkinson, p. 468 item 5 and Burden & Faires,
Theorem 7.3, p. 426–427). The only two “twists” with which we shall have to be concerned are (1): because
the right-hand vector in an inner product gets its components complex-conjugated, for complex scalars \( \mu \) we
shall have

\[
\langle z, (\mu w) \rangle = \overline{\mu} \cdot \langle z, w \rangle,
\]

although scalar multiples of the left-hand vector “come out through the inner product” without getting
conjugated, and (2): for the same reason, for any \( n \times n \) complex matrix \( A \)

\[
\langle Az, w \rangle = w^* Az = (A^* w)^* z = \langle z, A^* w \rangle.
\]

It follows that the matrices satisfying \( A = A^* \)—the **self-adjoint**, **conjugate-symmetric** or **Hermitean**
matrices—will have the important special property that they cross the inner product without changing:

\[
\langle Az, w \rangle = \langle z, A w \rangle.
\]

\(^{(3)}\) Unless we are physicists, in which case we introduce the same twist, but in the first factor rather than the second. Those who
wish to follow that convention in their other work will have to read what follows mutatis mutandis.
Of course if the entries in \( A \) are real, then \( A = A^\ast \) if and only if \( A = A^T \); thus we find out what real symmetric matrices do as a byproduct of our investigations into self-adjoint complex matrices.

As a sample of what makes self-adjoint matrices special, here is an easy proposition. Note that we have to use complex scalars here even though after we’re done we will have found out that a real symmetric matrix can have only real eigenvalues.

**Proposition:** Let \( A \) be a complex \( n \times n \) matrix satisfying \( A = A^\ast \) (so in particular, it could be a real matrix satisfying \( A = A^T \)). Then (1) every complex eigenvalue of \( A \) is in fact real, and (2) if \( \lambda \neq \mu \) are distinct eigenvalues of \( A \), then each eigenvector of \( A \) belonging to \( \lambda \) is perpendicular to each eigenvector of \( A \) belonging to \( \mu \).

**Proof.** Of (1): Let \( z \) be an eigenvector of \( A \) belonging to the eigenvalue \( \lambda \). Then we have

\[
\lambda \langle z, z \rangle = \langle (\lambda z), z \rangle = \langle Az, z \rangle = \langle z, Aw \rangle = \langle z, (\lambda w) \rangle = \overline{\lambda} \langle z, z \rangle.
\]

Comparing sides gives \( \lambda \|z\|^2 = \overline{\lambda} \|z\|^2 \), or \( \|z\|^2 \cdot (\lambda - \overline{\lambda}) = 0 \). If \( z \neq 0 \) then we must have \( \lambda - \overline{\lambda} = 0 \), i.e., \( \lambda = \overline{\lambda} \), and a complex number is real if and only if it equals its complex conjugate (why?). Of (2): Let \( z \) be an eigenvector of \( A \) belonging to the eigenvalue \( \lambda \) and let \( w \) be an eigenvector of \( A \) belonging to the eigenvalue \( \mu \) (we now know that both of those eigenvalues must be real numbers). Then

\[
\lambda \langle z, w \rangle = \langle (\lambda z), w \rangle = \langle Az, w \rangle = \langle z, Aw \rangle = \langle z, (\mu w) \rangle = \overline{\mu} \langle z, w \rangle.
\]

The conjugation bar on \( \overline{\mu} \) has no effect, because—as we already know—the eigenvalues of \( A \) must be real numbers. The equation above is therefore equivalent to \( (\lambda - \mu) \langle z, w \rangle = 0 \). If \( \lambda \neq \mu \) then the first factor is not zero, so one must have \( \langle z, w \rangle = 0 \), i.e., \( z \perp w \).

It can be shown that a self-adjoint matrix \( A \) always has an orthonormal basis of eigenvectors, i.e., that there is a set of \( n \) vectors \( \{ q_j : j = 1, \ldots, n \} \) satisfying the relation

\[
\langle q_j, q_k \rangle = \begin{cases} 1 & \text{if } j = k ; \\ 0 & \text{if } j \neq k \end{cases},
\]

such that each \( q_j \) is an eigenvector of \( A \) belonging to some necessarily real eigenvalue \( \lambda_j \) (the eigenvalues may not be distinct among themselves). In words, the \( q_j \)’s are all of unit \( \ell^2 \)-norm length and perpendicular to one another. This is the Principal Axes Theorem 7.4 of Atkinson, p. 476 ff., to which we refer for the proof and which (for the moment) we’ll take for granted. Our notation is slightly different since his \( u^{(j)} \) is our \( q_j \), so what you see here looks like the case in which all the scalars are real. The “multiplication table” for such basis vectors has an appealing matrix form: if \( Q \) is the matrix whose columns are the vectors \( q_j \) in order, then

\[
Q^\ast Q = \begin{bmatrix} q_1^\ast \cdots q_n^\ast \end{bmatrix} \begin{bmatrix} q_1 \cdots q_n \end{bmatrix} = \begin{bmatrix} \lambda_1 \cdots \\ \vdots \\ \lambda_n \end{bmatrix} \begin{bmatrix} \lambda_1 q_1 \cdots \lambda_n q_n \end{bmatrix} = \mathbf{I}_n.
\]

Thus \( Q^\ast = Q^{-1} \). (Note that consequently, because \( QQ^{-1} = \mathbf{I} \) is also true, we have \( QQ^\ast = \mathbf{I} \).) As a matrix whose columns are eigenvectors of \( A \), \( Q \) diagonalizes \( A \):

\[
Q^\ast AQ = Q^\ast A \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix} = Q^\ast \begin{bmatrix} \lambda_1 q_1 & \cdots & \lambda_n q_n \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.
\]

\[\text{(\ast)}\]
Thus $A$ “looks like a diagonal matrix,” and that resemblance is a very special one: it does not change the natural norm $\|A\|_2$. The reason is that the property $Q^* = Q^{-1}$ characterizes matrices that leave the inner product, and therefore the $l^2$ norm, unchanged: since $\langle Qz, Qw \rangle = \langle Q^*Qz, w \rangle$ in all cases, if $Q^* = Q^{-1}$ then we have $Q^*Q = I$ so $\langle Qz, Qw \rangle = Z \langle z, w \rangle$ for any two vectors, while if that equation holds for all pairs of vectors then $\langle Q^*Qz, w \rangle = \langle z, w \rangle$ holds for all pairs of vectors, which would be impossible unless $Q^*Q = I$ held for every $z$, i.e., unless $QQ^* = I$ held. So that condition certainly gives $\|Qz\|_2 = \sqrt{\langle Qz, Qz \rangle} = \|z\|_2$ for every $z$ in $\mathbb{R}^n$ or $\mathbb{C}^n$. Returning to the equation (*) above, we see that if we give the usual name $\Lambda$ to the diagonal matrix,

$$
\Lambda = \begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_n
\end{bmatrix},
$$

then $Q^*AQ = \Lambda$ means the same thing as $QAQ^* = \Lambda$, and so

$$
\|\Lambda\|_2 = \max \{ \|\Lambda z\|_2 : \|z\|_2 \leq 1 \}
= \max \{ \|QAz\|_2 : \|z\|_2 \leq 1 \} = \max \{ \|Q\Lambda z\|_2 : \|z\|_2 \leq 1 \} = \|A\|_2.
$$

Now it is very easy to see what the natural $l^p$-norm of a diagonal matrix $A$ is, for any $1 \leq p < \infty$ (the proof for $p = \infty$ is similar and—if anything—easier): if $z = (z_1, \ldots, z_n)^T$ and $\|z\|_p \leq 1$ then

$$
\|\Lambda z\|_p = \left( \sum_{j=1}^n |\lambda_j z_j|^p \right)^{\frac{1}{p}}
\leq \left( \sum_{1 \leq j \leq n} |\lambda_j|^p \right)^{\frac{1}{p}} \left( \sum_{j=1}^n |z_j|^p \right)^{\frac{1}{p}} \leq \max_{1 \leq j \leq n} |\lambda_j|,
$$

but on the other hand if one chooses $z = e_k$ where $|\lambda_k| = \max_{1 \leq j \leq n} |\lambda_j|$ then one gets $\|\Lambda e_k\|_p = |\lambda_k|$, so in fact one has exactly $\|\Lambda\|_p = \max_{1 \leq j \leq n} |\lambda_j|$. Remember this all started with a self-adjoint complex (or real symmetric) matrix $A$: so we can now assert the

**Proposition:** If $A = A^*$ is a real symmetric or self-adjoint complex matrix, then $\|A\|_2 = \max \{|\lambda_j| : 1 \leq j \leq n\}$, where the $\lambda_j$’s are the (real) eigenvalues of $A$.

Extracting this number is a much more complicated computational matter than the simple computation for $\|A\|_\infty$ given by the Geršgorin expression (Atkinson’s formula (7.3.17), Burden & Faires’ Theorem 7.11), but this is the truth so we’re stuck with it. Let us note for future reference (in computing condition numbers) that since it is fairly obvious that

$$
\Lambda^{-1} = \begin{bmatrix}
\lambda_1^{-1} & 0 & \cdots & 0 \\
0 & \lambda_2^{-1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_n^{-1}
\end{bmatrix}
$$

and that $A^{-1} = QA^{-1}Q^*$, we must also have $\|A^{-1}\|_2 = 1/(\min \{|\lambda_j| : 1 \leq j \leq n\})$.

We can pass from knowing the natural $l^2$-norm of self-adjoint matrices to knowing the natural $l^2$-norm of general matrices by using the following simple fact:
Proposition: \( \|A\|_2^2 = \|A^*A\|_2 \) for any \( n \times n \) matrix \( A \).

Proof. One always has \( \|A\|_2 = \|A\|_2 \), because if \( w \) is a vector with \( \|w\|_2 \leq 1 \) for which \( \|Aw\|_2 = \max \{ \|Az\|_2 : \|z\|_2 \leq 1 \} = \|A\|_2 \), then dotting with a unit vector \( u \) pointing in the direction of \( Aw \) gives us
\[
\|A\|_2 = \langle Aw, u \rangle = \langle w, A^*u \rangle \\
\leq \|w\|_2 \|A^*u\|_2 \leq \|A\|_2
\]
where the “\( \leq \)" at the beginning of the second line is the Cauchy-Schwarz inequality. This gives \( \|A\|_2 \leq \|A^*\|_2 \), but applying it to \( A^* \) gives \( \|A^*\|_2 \leq \|A\|_2 \) also, so the two norms are equal. Now with no further analysis we would know \( \|A^*A\|_2 \leq \|A^*\|_2 \|A\|_2 = \|A\|_2^2 \), because the \( \ell^2 \) natural norm satisfies \( \|BC\|_2 \leq \|B\|_2 \|C\|_2 \), the same as any natural matrix norm. However, for any vector \( z \) with \( \|z\|_2 \leq 1 \) we can write
\[
\|Az\|_2^2 = \langle Az, Az \rangle = \langle A^*A, z \rangle \leq \|A^*A\|_2 \cdot 1
\]
(the “\( \leq \)" is again the Cauchy-Schwarz inequality), so we also have \( \|A\|_2^2 \leq \|A^*A\|_2 \) and the two inequalities together give the equality asserted by the proposition.

Now any matrix \( A^*A \) is automatically self-adjoint (or symmetric), since \( (A^*A)^* = (A^*)^*A^* = A^*A \).

It is not difficult to see that all the eigenvalues of such a matrix are nonnegative: if \( A^*Az = \lambda z \) and without loss of generality we take \( \|z\|_2 = 1 \), then \( \lambda = \langle Az, z \rangle = \langle A^*Az, z \rangle = \|Az\|_2^2 \geq 0 \). Thus the norm of such a matrix is automatically its largest eigenvalue—numbering the eigenvalues in order of decreasing size, we can call it \( \lambda_1 \).

Putting this all together, we have

Proposition: The natural \( \ell^2 \)-norm of an \( n \times n \) matrix \( A \) is given by
\[
\|A\|_2 = \sqrt{\lambda_1}, \text{ where } \lambda_1 \text{ is the largest eigenvalue of } A^*A.
\]

Actual computation of this number by hand can be fairly unpleasant—Atkinson offers the example on p. 489—but the fact that it can be expressed in this way can be very useful for thinking about the \( \ell^2 \)-norm size of vectors. Looking forward to the notion of condition numbers again, we would like a similar expression for \( \|A^{-1}\|_2 \). It seems reasonable that one should have
\[
\|A^{-1}\|_2 = 1/\sqrt{\lambda_n}, \text{ where } \lambda_n \text{ is the smallest eigenvalue of } A^*A.
\]
and in fact this is true, but establishing it is a bit more subtle than one might at first suppose. The problem is that both the star operation and the operation of taking inverses reverse the order of multiplication, so that while \( (A^{-1})^* = (A^*)^{-1} \) (why?), one gets \( (A^{-1})^*A^{-1} = (A^*)^{-1}A^{-1} = (AA^*)^{-1} \). Thus the self-adjoint matrix one has to look at to find the norm of \( A^{-1} \) is not the same as the inverse of the matrix one looked at to find the norm of \( A \). However, these matrices have the same eigenvalues, because of the following little

Proposition: If \( B \) and \( C \) are two \( n \times n \) matrices, then every eigenvalue of \( BC \) is an eigenvalue of \( CB \) (and vice versa).

Proof. First of all, since \( \det BC = \det CB \) one of these products will have zero as an eigenvalue (i.e., be singular) if and only if the other product has zero as an eigenvalue. To handle nonzero eigenvalues \( \lambda \) of \( BC \), suppose \( BCx = \lambda x \) where \( x \) is a nonzero eigenvector belonging to \( \lambda \), and let \( y = Cx \neq 0 \) (if it were zero then \( \lambda \) would have been zero). Then \( CBy = CBCx = C(\lambda x) = \lambda Cx = \lambda y \), so \( \lambda \) is an eigenvalue of \( CB \) (with eigenvector \( y \)). Thus every eigenvalue of \( BC \) is an eigenvalue of \( CB \), and since the situation is symmetric in \( B \) and \( C \) we have proved the lemma.\(^{(4)}\)

\(^{(4)}\) This statement is “true with an exception” even when the two matrices \( B \) and \( C \) are rectangular, but of such sizes (one \( m \times n \), the other \( n \times m \)) that both products make sense. The exception is that zero may be an eigenvalue of the product in one order but not in the other. Try to find an example. People who have done a little functional analysis might want to try the following: let \( B : X \rightarrow Y \) and \( C : Y \rightarrow X \) be (bounded) linear mappings between Banach spaces. Express the resolvent \( (A - BC)^{-1} \) in terms of the resolvent \( (A - CB)^{-1} \) under the assumption that \( \lambda \neq 0 \). That will show that the resolvent sets of the two operators are the same (excluding zero). Thus eigenvalues and eigenvectors have nothing to do with the case: it’s all just spectral theory.
It follows that the eigenvalues of $A^*A$ are the same as those of $AA^*$, and thus so are the eigenvalues of their inverses; so the norm of $A^{-1}$ is the reciprocal of $\sqrt{\lambda_n}$, where $\lambda_n$ is the smallest eigenvalue of $A^*A$. As we shall see, the condition number of a matrix is defined to be $\|A\| \cdot \|A^{-1}\|$, so in the $\ell^2$ natural norm, the condition number of $A$ will be $\sqrt{\lambda_1/\lambda_n}$, where the $\lambda$s are the largest and smallest eigenvalues of the matrix $A^*A$. As we shall see in the next §, this number can also be described as the quotient of the largest singular value of $A$ by its smallest singular value.

8. The Singular Value Decomposition. We are so close to having the SVD that it would be unfortunate not to produce it at this point. Suppose that $A$ is an $m \times n$ matrix, not necessarily square. Then the product $A^*A$ still makes sense—the dimensions are right—and is an $n \times n$ (square) matrix. It is easy to check that, even though $A$ may not have been square, we still have the relations

1. $Ax = 0$ if and only if $A^*Ax = 0$: on the one hand $Ax = 0$ obviously implies $A^*Ax = Ax^*A = 0 = 0$, while on the other hand if $A^*Ax = 0$ then $0 = \langle 0, x \rangle = \langle A^*Ax, x \rangle = \langle Ax, Ax \rangle = \|Ax\|^2_2$, so one must have $Ax = 0$ (since it is a vector of length zero).

2. $\|A\|^2 = \|A^*A\|_2$. The proof given above did not require that $A$ be square.

3. All the eigenvalues of $A^*A$ are nonnegative. The proof given above did not require that $A$ be square.

4. The natural $\ell^2$-norm of $A$ is given by $\|A\|_2 = \sqrt{\lambda_1}$, where $\lambda_1$ is the largest eigenvalue of $A^*A$. Again, the proof given above did not require that $A$ be square.

Now because $A^*A$ is square ($n \times n$ when $A$ is $m \times n$) and self-adjoint in all cases, the diagonalization results that we discussed above apply to $A^*A$. We can thus find an orthonormal basis of $\mathbb{R}^n$ or $\mathbb{C}^n$ consisting of eigenvectors of $A^*A$, that is (again), a set of $n$ vectors $\{q_j : j = 1 \ldots, n\}$ satisfying the relation

$$\langle q_j, q_k \rangle = \begin{cases} 1 & \text{if } j = k; \\ 0 & \text{if } j \neq k \end{cases},$$

such that each $q_j$ is an eigenvector of $A^*A$ belonging to some necessarily nonnegative eigenvalue $\lambda_j$ (again, the eigenvalues may not be distinct among themselves). Without loss of generality, we may assume that the $q_j$’s have been indexed so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0 = \lambda_{r+1} = \cdots$, with the obvious changes if $\lambda_1 = 0$ (which would only happen if $A$ was the zero matrix) or $r = n$ (so that all the eigenvalues of $A^*A$ were strictly positive).(5) The “multiplication table” for such basis vectors again takes the form that if $Q$ is the matrix whose columns are the vectors $q_j$ in order, then

$$Q^*Q = [q_i^* q_j] = [\langle q_i, q_j \rangle] = I_n,$$

so again $Q^* = Q^{-1}$, and $Q$ diagonalizes $A^*A$:

$$Q^*(A^*A)Q = Q^*(A^*A) \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix} = Q^* \begin{bmatrix} \lambda_1 q_1 & \cdots & \lambda_n q_n \\ \vdots & \ddots & \vdots \\ \lambda_1 & \cdots & \lambda_n \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.$$  

---

(5) The number $r$ of nonzero eigenvalues will turn out to be the rank of the matrix $A$, the dimension of the subspace of $m$-dimensional space spanned by its columns.
The vectors \( \{q_j : j = 1, \ldots, n\} \) are in general not eigenvectors of \( A \), since in general they may be \( n \)-dimensional vectors while each \( Aq_j \) belongs to \( m \)-dimensional space. However, they are orthogonal to each other in \( m \)-dimensional space: we have for all \( 1 \leq i, j \leq n \)

\[
(Aq_i, Aq_j) = (A^*Aq_i, Aq_j) = (\lambda_i q_i, q_j) = \lambda_i (q_i, q_j) = \lambda_i \delta_{ij} = \begin{cases} \lambda_i & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}.
\]

It follows that if we set \( \mu_j = \sqrt{\lambda_j} \) for \( \lambda_j \neq 0 \), then the set of \( r \) vectors \( \{v_j = \frac{1}{\mu_j} Aq_j\}_{j=1}^r \) is orthonormal, i.e., obeys the “multiplication rule” \( \langle v_j, v_k \rangle = \delta_{jk} \) for \( 1 \leq j, k \leq r \). At this point we can already see that the \( v_j \)'s form an orthonormal basis for the column space of \( A \): on one hand, as an orthonormal set, they are automatically linearly independent; \(^{(6)}\) but on the other they automatically span the column space of \( A \), because that is the space of all vectors in \( m \)-dimensional space that have the form \( Ax \) for some \( x \) in \( n \)-dimensional space, and if we write out \( x \) in terms of the \( q_j \)'s:

\[
x = \langle x, q_1 \rangle q_1 + \cdots + \langle x, q_n \rangle q_n
\]

and apply \( A \), we get

\[
Ax = A([x, q_1] q_1 + \cdots + [x, q_n] q_n) = [x, q_1] Aq_1 + \cdots + [x, q_n] Aq_n = \langle x, q_1 \rangle v_1 + \cdots + \langle x, q_r \rangle v_r
\]

because \( Aq_j = 0 \) for \( j \geq r \), if there are any such \( q_j \)'s. Thus every vector in the column space of \( A \) is a linear combination of the \( \{v_j\}_{j=1}^r \). It follows that \( r \) is the dimension of the column space of \( A \), and since that dimension is by definition the rank of \( A \), we have proved the assertion that the number \( r \) of nonzero eigenvalues of \( A^*A \), or equivalently the number of nonzero vectors \( Aq_j \), is in fact equal to the rank of \( A \).

In all cases we must have \( r \leq m \), since the dimension of a subspace of \( m \)-dimensional space—e.g., the column space of \( A \)—can be at most \( m \). If \( r < m \), one may choose (unfortunately, in a pretty arbitrary manner) \( m - r \) additional vectors \( \{v_j\}_{j=r+1}^m \) of unit length, orthogonal to all the \( \{v_j\}_{j=1}^r \), and also pairwise orthogonal to each other, such that \( \{v_j\}_{j=1}^m \) is an orthonormal basis of \( m \)-dimensional space. It then obeys the dot-product rule \( \langle v_j, v_k \rangle = \delta_{jk} \), which again means that the \( m \times m \) matrix

\[
V = \begin{bmatrix}
\vdots & \vdots & \cdots & \vdots \\
v_1 & v_2 & \cdots & v_m \\
\vdots & \vdots & \cdots & \vdots
\end{bmatrix}
\]

whose columns are the \( \{v_j\}_{j=1}^m \) has the property that its transpose is its inverse:

\[
V^*V = [v_i^* v_j] = [\langle v_i, v_j \rangle] = I_m.
\]

We can now figure out what the matrix \( F = V^*AQ \) is by finding its columns. If \( e_{j,n} \) denotes the \( j \)-th of the “standard basis vectors” \( e_{j,n} = (0, \ldots, 1, \ldots, 0)^T \) of \( n \)-dimensional space, then for any \( m \times n \) matrix \( B \) the \( j \)-th column of \( B \) is just \( Be_j \). \( j = 1, \ldots, n \). So since \( Fe_j = V^*AQe_j = V^*A[e_j] = V^*\mu_jv_j = \mu_j(0, \ldots, 1, \ldots, 0)^T = \mu_j e_{j,m} \) where \( e_{j,m} \) denotes the \( j \)-th of the “standard basis vectors” \( e_{j,m} = (0, \ldots, 1, \ldots, 0)^T \) of \( m \)-dimensional space. But of course that says that \( F \) is a (possibly “incomplete”) diagonal matrix with the \( \mu_j \)'s running down its main diagonal for \( r \) nonzero entries:

\[
F = \begin{bmatrix}
\mu_1 & 0 & \cdots & \cdots & \cdots \\
0 & \mu_2 & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & \mu_r & \cdots \\
0 & 0 & \cdots & 0 & \cdots
\end{bmatrix}.
\]

\(^{(6)}\) This is a good point at which to review the reason that this statement is true, if it does not come to you immediately.
And now, since \( V^* = V^{-1} \) and \( Q^* = Q^{-1} \), we have

\[
A = V(V^*A)QQ^* = VFQ^* = V \begin{bmatrix}
\mu_1 & 0 & \cdots & 0 \\
0 & \mu_2 & \cdots & 0 \\
0 & 0 & \ddots & \vdots \\
0 & 0 & \cdots & \mu_r \\
0 & 0 & \cdots & 0
\end{bmatrix} Q^*.
\]

Such a factorization of \( A \) is called a **Singular Value Decomposition** (or SVD for short) of the matrix \( A \); the \( \mu_j \)'s are called the **singular values** of \( A \). The factorization is not quite unique: we saw that if \( r < m \) then there would be some arbitrariness in the choice of \( V \), and similarly that there is some arbitrariness in the choice of our \( Q \) (which, again, is Atkinson’s \( U \)). However, just the existence of the SVD is very handy. Geometrically, the SVD says that the action of any (even non-square) matrix \( A \) can be described as follows: rotate \( n \)-dimensional space (by \( Q^* \)) in such a way as to carry the orthonormal basis of \( n \)-dimensional space formed by the eigenvectors of \( A^*A \) into the standard basis; send the \( j \)-th standard basis vector of \( n \)-dimensional space to \( \mu_j \)-times the \( j \)-th standard basis vector of \( m \)-dimensional space, \( j = 1, \ldots, r \); finally, rotate \( m \)-dimensional space so that the vectors just described are rotated into the images under \( A \) (in \( m \)-dimensional space) of the orthonormal basis of \( n \)-dimensional space formed by the eigenvectors of \( A^*A \). This geometrical interpretation makes it easy to see that \( \|A\|_2 = \mu_1 \), the largest singular value: in the factorization \( A = VFQ^* \), if we feed in a vector \( x \) of length 1 from \( n \)-dimensional space, then \( Q^*x \) has the same length as \( x \), \( F \) multiplies the length of that vector by at most \( \mu_1 \)—and there is a vector whose length is multiplied by exactly \( \mu_1 \)—and then \( Ax = V[FQ^*x] \) has the same length as \( FQ^*x \). Thus \( A \) multiplies the length \( \|x\|_2 \) by at most \( \mu_1 \), and there is a vector whose length is multiplied by exactly that factor; so \( \|A\|_2 \) is exactly \( \mu_1 \).

Of course one would like algorithms for computing the SVD of a given matrix. They exist; they are related to the algorithms for finding the QR factorization of a given matrix, and we should look at that first. People who want to open their presents early can find a discussion of these algorithms in Golub & Van Loan, §8.3.2, pp. 430 ff. Matlab has a built-in function, called (logically enough) **svd**, for producing singular-value decompositions. {Note that the factorization \( QAQ^* = A \) derived above is itself a SVD of \( A \) when \( A \) is a positive definite symmetric matrix as defined in the next §. It follows that routines (e.g., Matlab’s svd) that produce SVDs will produce interesting results when applied to symmetric matrices, although if the symmetric matrix is positive definite this may not be the most efficient way to diagonalize it.}

### 9. Positive Definite Matrices

The considerations we just made help us to understand why the **positive definite** matrices defined in Atkinson’s problem 15, p. 499 and his discussion of Cholesky factorization on p. 524 (see also Burden & Faires’ Def. 6.21, p. 406) arise naturally and have considerable importance. Since these matrices are symmetric, they can be diagonalized by a matrix \( Q \) for which \( Q^* = Q^{-1} \), as on pp. 12–14 above, and it is pretty obvious that all inner products of the form \( \langle Ax, x \rangle = x^*Ax = x^*QAQ^*x = (Q^*x)^*AQ^*x = \langle A^*Q^*x, Q^*x \rangle \) are \( > 0 \) for \( x \neq 0 \) if and only if all the entries on the diagonal of the diagonal matrix \( A \) are strictly positive, which is basically the Atkinson problem on p. 499. (Cf. Burden & Faires’ Theorem 9.12, p. 544.) The fact that \( \langle A^*Ax, x \rangle = \langle Ax, Ax \rangle = \|Ax\|_2^2 \geq 0 \) with equality only if \( Ax = 0 \) also tells us that any matrix of the form \( A^*A \), where \( A \) is nonsingular, must be a positive definite matrix.

From a computational point of view, the important things about positive definite matrices are the following, presented by Burden & Faires on pp. 407–412. The next proposition is their Theorem 6.22, at least in the case in which the matrix \( A \) is real.
Proposition: For positive definite (self-adjoint) $n \times n$ matrices $A$:

1. $A^{-1}$ exists;
2. Each diagonal entry $a_{jj} > 0$; more sharply, $|a_{jk}| \leq \max_{1 \leq j \leq n} a_{jj}$ holds for each entry $a_{jk}$ of $A$;
3. $|a_{jk}| < \sqrt{a_{jj}a_{kk}}$ holds for each $1 \leq j, k \leq n$.

Proof. Of (1): if $A$ were not invertible, then there would be a nonzero vector $x \in \mathbb{C}^n$ for which $Ax = 0$, but then one would have $0 = \langle Ax, x \rangle$ contrary to the definition of positive-definiteness ($\langle Ax, x \rangle > 0$ must hold whenever $x \neq 0$). Of the first assertion of (2): take $x = e_j$, the $j$-th standard basis vector, and then $0 < \langle Ae_j, e_j \rangle = a_{jj}$. Of (3): let $\rho e^{i\theta}$ be a complex number written in polar form (but allowing $\rho$ to be negative if we wish) and let $x = \rho e^{i\theta} e_j + e_k$. Plugging this into the definition of positive definiteness, we get (as a moment’s computation will show)

$$0 < \langle A[\rho e^{i\theta} e_j + e_k], [\rho e^{i\theta} e_j + e_k] \rangle = \rho^2 a_{jj} + \rho[\rho e^{i\theta} a_{jk} + e^{-i\theta} a_{kj}] + a_{kk}$$

$$= a_{jj} \rho^2 + 2\Re[e^{i\theta} a_{jk}] \rho + a_{kk}$$

because $e^{i\theta} a_{jk}$ and $e^{-i\theta} a_{kj}$ are conjugates.

This expression has real coefficients and is quadratic in the real variable $\rho$, and its value is positive for all values of $\rho \in \mathbb{R}$. This is possible only if its discriminant ("$b^2 - 4ac$") is negative, and therefore

$$4\{\Re[e^{i\theta} a_{jk}]\}^2 - 4a_{jj}a_{kk} < 0$$

$$(\Re[e^{i\theta} a_{jk}])^2 < a_{jj}a_{kk}.$$ But $e^{i\theta}$ is still in our hands, and if we had chosen it so that $|a_{jk}| = \Re[e^{i\theta} a_{jk}]$, we would have

$$|a_{jk}|^2 < a_{jj}a_{kk}$$

which is obviously equivalent to (3). The second assertion of (2) now follows from (3), because for any $j, k$

$$|a_{jk}| < \sqrt{a_{jj}a_{kk}} \leq \max_{1 \leq j \leq n} a_{jj}$$

where the second inequality is obvious.

For another useful result (Burden & Faires’ Theorem 6.25, p. 409, and its corollaries 6.26 and 6.27), let $A$ be a real positive definite (symmetric) matrix, and consider its leading principal diagonal submatrices (as defined on p. 5 above). It is easy to see that these are also positive definite: they inherit their symmetry (or conjugate symmetry) from $A$, and they inherit their positivity as follows. Given $0 \neq z = (z_1, \ldots, z_k)^T \in \mathbb{C}^k$ we may form $0 \neq x = (z_1, \ldots, z_k, 0, \ldots, 0)^T \in \mathbb{C}^n$. By the definition of matrix multiplication, we will have

$$\langle A_k z, z \rangle = z^* A_k z = x^* A x = \langle Ax, x \rangle > 0$$

and since $0 \neq z \in \mathbb{C}^k$ was arbitrary, this shows that $A_k$ is positive definite and therefore nonsingular. The Principal Axes Theorem on p. 13 above implies that the determinant of any positive definite matrix is positive, since (in the notation of the theorem)

$$\det A = \det[QAQ^*] = (\det Q)(\det A)(\det Q^{-1}) = \det A = \lambda_1 \cdots \lambda_n > 0.$$ Applying this to each $A_k$, we see that all the leading principal diagonal submatrices of $A$ have positive determinants (and are consequently invertible). The proposition on p. 5 above now assures us that Gaussian

---

(7) This is a “geometric mean” estimate of the off-diagonal elements. Curiously, Van Loan gives a less-sharp “arithmetic mean” estimate (without proof) on his p. 254.

(8) If one is working over the real scalars, the choices for $e^{i\theta}$ are just $\pm 1$. 

642:574:01 Spring 2000
elimination can be performed on $A$ without row interchanges. So we can begin to see that there are many characterizations of positive definite (symmetric) matrices. A few of the most useful ones are the following.

**Proposition:** The following properties of a real (square) matrix $A$ are logically equivalent:

(1) $A$ is positive definite symmetric;

(2) $A$ is symmetric, and Gaussian elimination for (all) linear systems $Ax = b$ can be performed with no row interchanges and with all pivots positive;

(3) $A$ can be factored as $LDL^T$, where $L$ is lower-triangular with ones on its diagonal and $D$ is diagonal with positive diagonal entries;

(4) $A$ can be factored as $LL^T$, where $L$ is lower-triangular with positive diagonal entries.

**Proof.** The discussion preceding the statement of the proposition contained a proof that (1) $\Rightarrow$ (2). If we assume (2) then Gaussian elimination = factorization of $A$ without row interchanges leads to a factorization $A = LU = LDV$ where $L$ is lower-triangular with $1$’s on the diagonal, $D$ is diagonal, and $V$ is upper-triangular with $1$’s on the diagonal. The diagonal elements of $D$ are the pivots in Gaussian elimination, which are by hypothesis positive. Since $A = A^T = (LDV)^T = V^TDL^T$ and the factorization is unique (see pp. 7–8 above), we must have $L^T = V$ and $A = LDL^T$; so we have shown that (2) $\Rightarrow$ (3). (3) $\Rightarrow$ (4) is rather trivial: one simply writes $LDL^T = (L\sqrt{D})(L\sqrt{D})^T$ where $\sqrt{D}$ is the diagonal matrix whose $ii$-position diagonal entry is $\sqrt{d_{ii}}$ (positive square roots being taken as usual). Finally, (4) or indeed (3) implies (1): we can begin by observing that for real matrices $A$ (for which $A^T$ and $A^*$ mean the same thing), if we can write $A = LDL^T$ with all the entries of the diagonal matrix $D$ positive, then for any $0 \neq x \in \mathbb{R}^n$ we have

$$
\langle Ax, x \rangle = x^TAx = x^TLDL^Tx = (L^Tx)^TD(L^Tx)
$$

$$
= \sum_{i=1}^{n} d_{ii}y_i^2 \quad \text{where } L^Tx = y = (y_1, \ldots, y_n)^T
$$

$$
> 0 \quad \text{if } x \neq 0
$$

and so $A$ is positive definite.

This proposition contains Burden & Faires’ Thm. 6.25 and its corollaries 6.26 and 6.27. {It is also possible to characterize positive definite symmetric matrices as those symmetric matrices $A$ with the property that the determinants of all their leading principal submatrices $A_k$ are positive; such “determinantal” characterizations are somewhat old-fashioned and not well adapted to machine computation, though they’re occasionally handy for thinking purposes. The interested reader will find in the discussion above all the raw materials needed to prove that this characterization is valid.}

Actual computation of these $LDL^T$ and $LL^T$ factorizations can be made somewhat more efficient than ordinary Gaussian elimination by taking advantage of the symmetry of the matrix and of the fact that pivoting will not be necessary. The relevant algorithms are usually called **Cholesky factorization** algorithms. Atkinson discusses these on pp. 524–527; Burden & Faires’ Algorithms 6.5 and 6.6, pp. 410–412, give you computational details in a “classical” setting. For an intensive introductory discussion (in a Matlab context) of the algorithms of Cholesky factorization, with examples, it would be hard to do better than Van Loan’s §7.3 and §7.4, p. 254 ff., since this is exactly a part of Van Loan’s field of special interest. These §§ also contain a discussion of positive definiteness generally, including an example (boundary-value problem for a [discretized] Sturm-Liouville differential operator) showing where these matrices arise in nature.

**10. Error Analysis in Computational Linear Algebra.** The time constraints of this course make it impossible to do a detailed analysis of the errors that floating-point machine computation introduces into linear algebra. The philosophy of error analysis can be stated briefly, however: an attempt to solve $Ax = b$ on a computer will generally result in a solution of $\tilde{A}\tilde{x} = \tilde{b}$ instead: *i.e.*, one will get an exact solution $\tilde{x}$ of an equation whose matrix of coefficients $\tilde{A}$ contains errors. The true matrix of coefficients $A$ therefore has the form $A = A + E$, where $E$ is a matrix whose entries are the errors in the corresponding entries of $A$. 
If \( x = A^{-1}b \) is the true solution and \( \bar{x} \) the computed solution of \( Ax = b \), then, we have for any norm on vectors and its corresponding natural norm on matrices (= “associated matrix norm,” etc.)

\[
x = A^{-1}b = A^{-1}A\bar{x} = A^{-1}(A - E)\bar{x} = \bar{x} - (A^{-1}E)\bar{x},
\]

so

\[
||\bar{x} - x|| \leq ||A^{-1}|| ||E|| ||x|| = \frac{||E||}{||A||} ||A|| ||A^{-1}|| ||\bar{x}||.
\]

The factor \( ||A|| ||A^{-1}|| \) thus appears as a “maximum amplification factor for relative error in computed solutions”; it is the condition number of the matrix \( A \), which various authors denote by \( \text{cond}(A) \) (Atkinson’s choice, p. 530), \( K(A) \) (Burden & Faires’ choice), or \( \kappa(A)(\text{Kincaid & Cheney’s notation, as well as Van Loan’s; the latter author subscripts the} \ k \text{to show what \( \ell^p \)-norm he’s using, e.g.,} \kappa_1(A) \text{is the condition number of} \ A \text{in the} \ell^1\text{-norm). The last set-off inequality above can then be rewritten as}

\[
\frac{||e||}{||x||} = \frac{||x - \bar{x}||}{||x||} \leq \text{cond}(A) \frac{||E||}{||A||} \quad \text{(*)}
\]

where \( e = \bar{x} - x \) is the (vector) error in the computed value of \( x \). Thus, e.g., if the true coefficients of \( A \) are only known to a relative accuracy of \( 10^{-r} \), or if relative inaccuracy in \( \bar{A} \) of that order can be expected, and \( \text{cond}(A) \) is on the order of \( 10^s \), then it will be impossible to attain relative accuracy of the computed solution better than to within about \( 10^{-(r-s)} \).

As a sample of what to expect in the size of the error term \( E \), let me cite the results of two analyses. The first is found in S. D. Conte and C. de Boor, Elementary Numerical Analysis: an Algorithmic Approach, 3rd ed., McGraw-Hill, 1980, p. 179. (The proof, while not difficult, is long.) I’ll change the notation slightly to conform to Burden & Faires’ usage. For arbitrary matrices \( B = [b_{ij}] \) they let \( |B| = [|b_{ij}|] \) denote the matrix whose entries are the absolute values of the corresponding entries of \( B \), so \( B \leq C \) means that \( |b_{ij}| \leq |c_{ij}| \) holds for every \( i,j \)-position. (Note that for the \( \ell^p \)-norms one has \( ||A|| = \| \| A \| \| \), so that inequalities will give norm estimates in these interesting cases.) “First-nonzero-entry” denotes the pivoting strategy in which, on the \( k \)-th pass, the first row at or below the \( k \)-th that has a nonzero entry in the \( k \)-th column is used as the pivot row (a decidedly minimal pivoting strategy). Then

**Theorem:** Suppose that Gaussian elimination with first-nonzero-entry pivoting is used on the \( n \times n \) matrix \( A \) to produce the computed factorization \( PA = \bar{L}\bar{U} \), followed by back-solution to produce the computed solution \( \bar{x} \) of the equation \( Ax = b \). Assume that computation is done in floating-point arithmetic with unit roundoff \( \delta \leq 0.01 \). Then \( \bar{x} \) exactly satisfies a perturbed equation

\[
(A + P^{-1}E)\bar{x} = b
\]

with

\[
|E| \leq \delta_n |PA| + \delta_n (3 + \delta_n) |\bar{L}| |\bar{U}|
\]

and

\[
\delta_n = n (1.01) \delta .
\]

This gives a bound on the size of the relative error of the computed solution that can be used in (*) above. A “more realistic estimate if partial pivoting is used,” according to Conte and de Boor, is

\[
|E| \leq |PA| n \delta, \quad \frac{||x - \bar{x}||}{||x||} \leq \text{cond}(A) n \delta
\]

although in fact there should be a small term in the error estimate that grows quadratically with \( n \).

Kincaid & Cheney offer an extensive analysis of roundoff error in Gaussian elimination in their §4.8, p. 262 ff. The two most interesting theorems are probably

**Theorem 1:** Let \( A = (a_{ij}) \) be an \( n \times n \) nonsingular matrix whose elements are machine numbers in a computer whose unit roundoff is \( \delta \). The Gaussian algorithm with row pivoting produces matrices \( \bar{L} \) and \( \bar{U} \) such that

\[
\bar{L}\bar{U} = A + E \quad \text{where} \quad |e_{ij}| \leq 2n\delta \max_{1 \leq i,j,k \leq n} |a_{ij}^{(k)}| .
\]
Theorem 5: Let the elements of $A$ and $b$ be machine numbers. If the Gaussian algorithm with row pivoting is used to solve $Ax = b$, then the computed solution $\hat{x}$ is the exact solution of a perturbed system

$$(A + F)\hat{x} = b$$

in which $|f_{ij}| \leq 10n^2 \delta \rho$.

Here $n$ is (as usual) the order of the matrix $A$, $\rho = \max_{1 \leq i,j,k \leq n} |a_{ij}^{(k)}|$, and $\delta$ is the machine’s unit roundoff error. It is assumed that $n\delta < \frac{1}{3}$.

11. Iterative Refinement. Another place in which the condition number of $A$ comes in handy is in estimating the relative error of a computed solution $\hat{x}$ of $Ax = b$ — however the computation may have been derived. The computation that proves Atkinson’s (8.4.4), which is equivalent to the relation set off immediately below (it is given here as found in Burden & Faires’ Theorem 7.27, p. 462), contains no hypotheses about the method used to find $\hat{x}$; one simply finds that

$$\frac{\|e\|}{\|x\|} = \frac{\|\hat{x} - \hat{x}\|}{\|x\|} \leq \text{cond}(A) \frac{\|r\|}{\|b\|},$$

where $r = b - A\hat{x}$ is a computable vector called the residual vector for $\hat{x}$. Its uses are many. First of all, one can use the residual vector to make approximations of the condition number of $A$: one can show (see the narrative of Burden & Faires, p. 464–466, or see G. E. Forsythe & C. B. Moler, Computer Solution of Linear Algebraic Systems, Prentice-Hall, 1967, pp. 49–51 for the gory details) that if $\hat{x}$ was obtained by using Gaussian elimination with $t$-digit arithmetic to solve $Ax = b$, then if one repeats the process and solves $Ay = r$ using double-precision $= 2t$-digit arithmetic, the number $(\|\hat{y}\|/\|\hat{x}\|) \cdot 10^t$ will be a fairly good approximation to $\text{cond}(A)$.

Secondly, while it is nice to have an approximation to $\text{cond}(A)$, it is even better to have a good solution to $Ax = b$. The vector $\hat{y}$ just found as a solution of $Ay = r$ can be treated as a correction term: if $\hat{x}$ was not too bad an approximation to the true solution of $Ax = b$, then $\hat{x} + \hat{y}$ will be a better one. This process can be repeated. To analyze what that will do, we need some notation, and I will use Burden & Faires’ rather ungrateful notation in their Example 4, p. 467 ff., because their example is accompanied by an algorithm that immediately precedes it.\(^9\) The true equation is $Ax = b$, of course, and its true solution is $x$. The approximate equation-solver will be called $\hat{A}^{-1}$: in most cases it is the process of Gaussian elimination and back-solution that computes the solution $z$ of $Az = c$ when $c$ is given. (It is the true inverse of some matrix $\hat{A}$ which is “close” to $A$ but not quite right.) Then

$$\hat{x}^{(1)} = \hat{A}^{-1}b$$

the computed solution of $Ax = b$

$$r^{(1)} = b - A\hat{x}^{(1)}$$

the first residual

$$\hat{y}^{(1)} = \hat{A}^{-1}r^{(1)}$$

the computed solution of $Ay = r^{(1)}$

$$\hat{x}^{(2)} = \hat{x}^{(1)} + \hat{y}^{(1)}$$

the second, improved solution

$$\cdots$$

$$r^{(k)} = b - A\hat{x}^{(k)}$$

the $k$-th residual

$$\hat{y}^{(k)} = \hat{A}^{-1}r^{(k)}$$

the computed solution of $Ay = r^{(k)}$

$$\hat{x}^{(k+1)} = \hat{x}^{(k)} + \hat{y}^{(k)}$$

the next improved solution.

This gives us a recursion formula for the error at the $k$-th step:

$$x - \hat{x}^{(k+1)} = x - \hat{x}^{(k)} - \hat{y}^{(k)} = x - \hat{x}^{(k)} - \hat{A}^{-1}r^{(k)}$$

$$= x - \hat{x}^{(k)} - \hat{A}^{-1}(b - A\hat{x}^{(k)}) = x - \hat{x}^{(k)} - \hat{A}^{-1}(Ax - A\hat{x}^{(k)})$$

$$= (I - \hat{A}^{-1}A)(x - \hat{x}^{(k)}).$$

\(^9\) This will mean that I will slightly “fight the book” by deviating from Atkinson’s notation, but all that really happens is that the indices will shift by one click from Atkinson’s: his initial residual is $r^{(0)}$, while Burden & Faires’ is $r^{(1)}$. 

22
It follows that \( \| x - \hat{x}^{(k)} \| \leq \| I - \hat{A}^{-1} A \|^{k-1} \| x - \hat{x}^{(1)} \| \), so one can hope for linear convergence of the iterates if \( \hat{A}^{-1} A \) is reasonably close to the identity (i.e., if \( \| I - \hat{A}^{-1} A \| < 1 \)). From a computational point of view, it makes no sense to iterate beyond the step at which the \( \ell^\infty \)-norm of the correction term \( \hat{y}^{(k)} \) is smaller than the number of significant digits in the floating-point arithmetic, which is exactly the stopping rule that Burden & Faires suggest in the middle of p. 467.

It is possible to make a rather crude estimate of \( \| I - \hat{A}^{-1} A \| \). The true matrix of coefficients \( A \) has the form \( A = \hat{A} + E \), where \( E \) is a matrix whose entries are the errors in the corresponding entries of \( \hat{A} \). We can thus write

\[
\hat{A}^{-1} A = (A - E)^{-1} A = [A(I - A^{-1} E)]^{-1} A = (I - A^{-1} E)^{-1}.
\]

It is not difficult to show that if the matrix \( B \) satisfies \( \| B \| < 1 \), then the formal geometric series (the C. Neumann series or resolvent series) with matrix terms on the r. h. side of

\[
(I - B)^{-1} = \sum_{j=0}^{\infty} B^j
\]

converges to a matrix which is indeed the matrix inverse of the l. h. side. (This fact is discussed at great length in Atkinson’s §7.4, particularly in Theorem 7.10 ff. on p. 491 ff. See also Kincaid & Cheney’s discussion of iterative refinement, §4.5, p. 211 ff.) For sufficiently small \( \| E \| \) we therefore have

\[
\hat{A}^{-1} A = (I - A^{-1} E)^{-1} = \sum_{j=0}^{\infty} (A^{-1} E)^j = I + \sum_{j=1}^{\infty} (A^{-1} E)^j.
\]

\[
I - \hat{A}^{-1} A = -\sum_{j=1}^{\infty} (A^{-1} E)^j.
\]

\[
\| I - \hat{A}^{-1} A \| \leq \sum_{j=1}^{\infty} \| A^{-1} \|^j \| E \|^j = \frac{\| A^{-1} \| \| E \|}{1 - \| A^{-1} \| \| E \|} = \frac{\text{cond}(A) (\| E \|/\| A \|)}{1 - \text{cond}(A) (\| E \|/\| A \|)}.
\]

To keep this estimate < 1 we need to keep

\[
\text{cond}(A) \frac{\| E \|}{\| A \|} < \frac{1}{2},
\]

\[
\frac{\| E \|}{\| A \|} < \frac{1}{2 \cdot \text{cond}(A)},
\]

i.e., the relative error in the matrix \( A + E \) whose inverse is actually being used must be kept smaller than half the reciprocal of the condition number of \( A \). Even if one accepts the “realistic” bound \( \| E \| \leq n \delta \| A \| \) for the error in Gaussian elimination, one sees that iterative refinement may not result in improved accuracy when the matrix of coefficients \( A \) is so ill-conditioned that \( \text{cond}(A) n \delta \geq 1/2 \). Indeed, examples can be given in which iteration only makes matters worse.

Some less-detailed discussion of iterative correction, with examples, can be found in Van Loan’s §6.4, p. 234 ff.