1. Read section 2.5 of Lay at least through the material in the box on p.146. Then do problem 31 on p.150. In part (a) use the command \([L,U]=\text{lu}(A)\). In part (b) use the backslash operator twice. (See equation (2) on p.134.)

MATLAB uses finite precision arithmetic, so there will be round-off error in most calculations. In order to minimize the effect of round-off error the Gaussian elimination process must be modified by introducing the concept of partial pivoting. (The definition is given on p.20 of Lay.) The MATLAB command \(A\backslash b\) implements Gaussian elimination with partial pivoting. What accuracy should one expect?

MATLAB carries about 15 digits so its basic calculations are very accurate, with errors about \(10^{-15}\). We can quantify this more precisely. The numbers that can be represented in MATLAB are called floating point numbers. There is a special number in MATLAB denoted by \(\text{eps}\), that is defined to be the distance from 1 to the next floating point number. The command \(\text{eps}\) will produce \(\text{eps}\) for the computer you are using; \(\text{eps}\) is approximately \(2.22e-16\). The significance of \(\text{eps}\) is that it determines the round-off error committed when a number is rounded: If a positive number \(a\) is rounded, the round-off error is approximately \(a \times \text{eps}\).

Consider the linear system
\[
Ax = b,
\]
and let \(x^*\) be the computed solution (the number produced by the computer). There are two measures of accuracy of the computed solution \(x^*\). The error in \(x^*\) is defined to be
\[
e = x - x^*,
\]
and the residual is defined to be
\[
r = b - Ax^*.
\]
The norm (size) of \(e\) is one measure of the accuracy of \(x^*\); if the norm of \(e\) is small, then \(x^*\) is close to \(x\). The second measure of accuracy of \(x^*\) is the norm of \(r\); if the norm of \(r\) is small, then \(x^*\) nearly satisfies the linear system.

There are two principles that indicate the accuracy of the solution \(x^*\) produced by Gaussian elimination with partial pivoting (the method used in \(A\backslash b\)). The first is that the norm of the residual is nearly always small; to be more precise,
\[
\text{Norm of } r \propto \text{Norm of } x \times \text{Norm of } A \times \text{eps},
\]
i.e., for a moderate sized problem the norm of the residual is approximately \(\text{eps}\). The second principle is that
\[
\text{Norm of } e \propto \text{Norm of } x \times \text{cond}(A) \times \text{eps},
\]
where \(\text{Cond}(A)\) is a number that measures how close \(A\) is to being singular. By the norm of a vector or a matrix we mean roughly speaking, the maximum of the absolute
values of the elements of the vector or the matrix, respectively. If \( A \) is nearly singular, then \( \text{Cond}(A) \) is large, and the error may be large. Matrices for which \( \text{cond}(A) \) is large are called *ill-conditioned*. If \( \text{Cond}(A) \approx 10^t \), then the error would be about \( 10^t \times \text{eps} \approx 10^{16-t} \), i.e., we would have lost approximately \( t \) digits of accuracy. \( \text{Cond}(A) \) can be found with the command \( \text{cond}(A) \). When \( A \backslash b \) is used, if the calculations indicate that \( A \) is ill-conditioned, a warning message is printed, and a number \( \text{RCOND} \) which is an approximation to \( 1/\text{cond}(A) \), is printed. The following two problems illustrate these principles. Note: type \texttt{format long} before you do the requested calculations.

2. The \( n \times n \) matrix \( H_n \) whose elements are given by \( h_{i,j} = 1/(i + j - 1) \) is very ill-conditioned if \( n \) is large; it is called the *Hilbert Matrix*. It can be generated in MATLAB with the command \texttt{hilb(n)}. Let \( A \) be the Hilbert matrix of size 11. Let \( x = \text{ones}(11,1) \), and let \( b = Ax \). Now solve the system \( Ax = b \), obtaining \( x^* \). Since we know \( x \) exactly we can compute \( e \). Do this and also compute \( r \). What are the norms of these vectors? Are the above principles satisfied for this example?

3. Let

\[
A = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
4 & 5 & 6 & 3 & -2 \\
7 & 8 & 9 & -3 & 3 \\
3 & 4 & -2 & -4 & 10 \\
1 & 2 & 3 & 4 & 6
\end{pmatrix}
\]

Find \( \text{cond}(A) \). Is \( A \) ill-conditioned? Solve \( Ax = b \), where \( b = (55, 34, 53, 39, 60)^T \). How small would you expect \( e \) and \( r \) to be? The exact solution is \( x = (1, 2, 3, 4, 5)^T \). Calculate \( e \) and \( r \). Are the above principles satisfied for this example?