**Motivation** To solve $Ax = b$, we can row reduce the augmented matrix

$$\begin{bmatrix} a_1 & a_2 & \ldots & a_i & \ldots & a_n \mid b \end{bmatrix} \sim \ldots \sim \begin{bmatrix} c_1 & c_2 & \ldots & c_i & \ldots & c_n \mid d \end{bmatrix}$$

When we get to $\begin{bmatrix} c_1 & c_2 & \ldots & c_i & \ldots & c_n \end{bmatrix}$ in an echelon form (this is the “forward” part of the row reduction process), then we can either

i) use “back substitution” to solve for $x$

ii) continue on with the second (“backward”) part of the row reduction process until $\begin{bmatrix} c_1 & c_2 & \ldots & c_i & \ldots & c_n \end{bmatrix}$ is in row reduced echelon form, at which point it's easy to read off the solutions of $Ax = b$.

A computer might take the first approach; the second might be better for hand calculations. But both i) and ii) take about the same number of arithmetic operations (see the first paragraph about “back substitution” in Sec. 1.2 (p. 19) of the textbook).

Whether we use i) or ii) to solve $Ax = b$, it is sometimes necessary in applications to solve a large system $Ax = b$ many times for the same $A$ but changing $b$ each time — perhaps a system with millions of variables and solving thousands of times! For example, read the description of the aircraft design problem at the beginning of Chapter 2. It's too inefficient to row reduce $\begin{bmatrix} a_1 & a_2 & \ldots & a_i & \ldots & a_n \mid b \end{bmatrix}$ each time, even if a computer is doing the work.

The general idea: the same EROs would be repeated to row reduce the coefficient matrix $\begin{bmatrix} a_1 & a_2 & \ldots & a_n \end{bmatrix}$ to an echelon form $U$ each time $b$ is changed. To avoid the repetitions, these necessary row operations can be “recorded” in a matrix. This results in a factorization of $A$ into the form $A = LU$. Sometimes we can choose the EROs so that $Z$ is a unit lower triangular matrix and, in that case, we will rename the matrix $Z$ as $L$ (for “lower”). The result is called an “LU decomposition” or “LU factorization” of $A$.

**Terminology:** A lower triangular matrix is a square matrix that has only 0 entries above the main diagonal, for example

$$\begin{bmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix}$$

; the matrix is unit lower triangular if it's lower triangular and has only 1’s on the diagonal, for example

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

. The definitions for upper triangular and unit upper triangular are the same except that words “above the main diagonal” are replaced by “below the main diagonal.”
Here's the more formal definition:

**The LU Decomposition** An \( LU \) decomposition of an \( m \times n \) matrix \( A \) is a factorization

\[
A = L \cdot U \quad \text{where} \quad \begin{cases} 
U \text{ is an echelon form of } A, \\
L \text{ is a square unit lower triangular matrix}
\end{cases}
\]

A couple of observations:

- \( U \) is an echelon form of \( A \), so \( U \) must be the same shape as \( A \) : \( m \times n \)
- \( L \) must be square \( m \times m \) so that all the matrices in \( A + LU \) have the right sizes. And since \( L \) is unit lower triangular, \( L \) must be invertible (why? how many pivot positions in the rref of \( L \)? Look at the examples above).
- If \( A \) and \( U \) happen to be square, then \( U \), because it's an echelon form, will automatically be upper triangular. But even when \( U \) is not square, \( U \) has only 0's below the leading entry in each row, so \( U \) still “resembles” an upper triangular matrix.

It's not always possible to find an \( LU \) decomposition for \( A \). But when it is possible, the number of steps to find \( L \) and \( U \) is not much different from the number of steps required to solve \( A\mathbf{x} = \mathbf{b} \) (one time!) by row reduction. And if we can factor \( A \) in this way, then it takes relatively few additional steps to solve \( A\mathbf{x} = \mathbf{b} \) for \( \mathbf{x} \) each time a new \( \mathbf{b} \) is chosen.

Why? If we can write \( A\mathbf{x} = LU\mathbf{x} = \mathbf{b} \), then we can substitute \( U\mathbf{x} = \mathbf{y} \). Then the original matrix equation \( A\mathbf{x} = \mathbf{b} \) is replaced by two new ones.

\[
\begin{cases} 
L\mathbf{y} = \mathbf{b} \\
\mathbf{y} = U\mathbf{x}
\end{cases} \quad (*)
\]

This is an improvement (as Example 1 will show) because:

i) it's easy to solve \( L\mathbf{y} = \mathbf{b} \) for \( \mathbf{y} \) because \( L \) is lower triangular; and

ii) after finding \( \mathbf{y} \), then solving \( U\mathbf{x} = \mathbf{y} \) is also easy, because \( U \) is already in echelon form

iii) \( L \) and \( U \) remain the same whenever we change \( \mathbf{b} \).

Look at the details in the following example.

**Example 1** Suppose \( A\mathbf{x} = \mathbf{b} \) is the equation

\[
\begin{bmatrix}
1 & -1 & 2 \\
2 & 1 & 0 \\
0 & 4 & 2
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}
\end{bmatrix}
= 
\begin{bmatrix}
2 \\
1 \\
-1
\end{bmatrix}
\]

Here is an \( LU \) factorization of \( A \) (don't worry for now about where it came from):

\[
\begin{bmatrix}
1 & -1 & 2 \\
2 & 1 & 0 \\
0 & 4 & 2
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
0 & 4 & 1
\end{bmatrix}
\begin{bmatrix}
1 & -1 & 2 \\
0 & 3 & -4 \\
0 & 0 & \frac{22}{3}
\end{bmatrix}
\]

This is an improvement (as Example 1 will show) because:

i) it's easy to solve \( L\mathbf{y} = \mathbf{b} \) for \( \mathbf{y} \) because \( L \) is lower triangular; and

ii) after finding \( \mathbf{y} \), then solving \( U\mathbf{x} = \mathbf{y} \) is also easy, because \( U \) is already in echelon form

iii) \( L \) and \( U \) remain the same whenever we change \( \mathbf{b} \).
To solve 
\[
\begin{bmatrix}
1 & -1 & 2 \\
2 & 1 & 0 \\
0 & 4 & 2
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
0 & \frac{4}{3} & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= 
\begin{bmatrix}
1 & -1 & 2 \\
0 & 3 & -4 \\
0 & 0 & \frac{22}{3}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= 
\begin{bmatrix}
2 \\
1 \\
-1
\end{bmatrix}.
\]

\[
A \mathbf{x} \quad = \quad \uparrow 
\begin{bmatrix}
L \\
& U
\end{bmatrix}
\quad \uparrow 
\begin{bmatrix}
\mathbf{x} \\
& \mathbf{b}
\end{bmatrix}
\]

i) substitute \( U \mathbf{x} = \mathbf{y} \) to get the two equations

\[
(*) \quad \begin{cases}
Ly = b \\
y = U \mathbf{x}
\end{cases}
\quad \text{that is,} \quad 
\begin{cases}
L \mathbf{y} = \begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
0 & \frac{4}{3} & 1
\end{bmatrix} \mathbf{y} = \begin{bmatrix}
2 \\
1 \\
-1
\end{bmatrix} \\
y = U \mathbf{x} = \begin{bmatrix}
1 & -1 & 2 \\
0 & 3 & -4 \\
0 & 0 & \frac{22}{3}
\end{bmatrix} \mathbf{x}
\end{cases}
\]

ii) First solve for \( \mathbf{y} \). To do this, we could further reduce \([L \ b]\) to row reduced echelon form, or we can use “forward substitution” — just like back substitution but starting with the top equation because that's where the simplest equation is. Either way we need only a few steps to get \( \mathbf{y} \) because \( L \) is lower triangular. Using forward substitution gives:

\[
\begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
0 & \frac{4}{3} & 1
\end{bmatrix} \mathbf{y} = \begin{bmatrix}
2 \\
1 \\
-1
\end{bmatrix}, \quad \text{so} \quad \begin{cases}
y_1 = 2 \\
y_2 = 1 - 2y_1 = -3 \\
y_3 = -1 - \frac{4}{3}y_2 = 3
\end{cases}
\]

so \( \mathbf{y} = \begin{bmatrix}
y_1 \\
y_2 \\
y_3
\end{bmatrix} = \begin{bmatrix}
2 \\
-3 \\
3
\end{bmatrix} \)

iii) Substitute \( \mathbf{y} \) into the second equation to get 
\[
\begin{bmatrix}
1 & -1 & 2 \\
0 & 3 & -4 \\
0 & 0 & \frac{22}{3}
\end{bmatrix} \mathbf{x} = \begin{bmatrix}
2 \\
-3 \\
3
\end{bmatrix}, \quad \text{which we can quickly solve with back substitution or further row reduction. Using back substitution we start at the bottom and work up:}
\]

\[
\begin{cases}
x_3 = 3 \left( \frac{3}{22} \right) = \frac{9}{22} \\
3x_2 = -3 + 4(x_3) = -\frac{15}{11}, \quad \text{so} \quad x_2 = -\frac{5}{11} \\
x_1 = 2 + x_2 - 2(x_3) = \frac{8}{11}
\end{cases}
\]

\[
\text{So} \quad \mathbf{x} = \begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix} = \begin{bmatrix}
\frac{8}{11} \\
-\frac{5}{11} \\
\frac{9}{22}
\end{bmatrix}
\]
**Example 2** For the matrix \( A = \begin{bmatrix} 1 & -1 & 2 \\ 2 & 1 & 0 \\ 0 & 4 & 2 \end{bmatrix} \) in Example 1, how could you actually find the LU decomposition?

First, row reduce \( A \) to an echelon form \( U \). (It turns out that we need to be careful about what EROs we use during the row reduction but for now, just follow along; the discussion that comes later will explain what caution is necessary.) Each elementary row operation that we use corresponds to left multiplication by an elementary matrix \( E_i \), and we record those matrices at each step.

\[
A = \begin{bmatrix} 1 & -1 & 2 \\ 2 & 1 & 0 \\ 0 & 4 & 2 \end{bmatrix} \sim \begin{bmatrix} 1 & -1 & 2 \\ 0 & 3 & -4 \\ 0 & 4 & 2 \end{bmatrix} \sim \begin{bmatrix} 1 & -1 & 2 \\ 0 & 3 & -4 \\ 0 & 0 & \frac{22}{3} \end{bmatrix} = U \text{ (an echelon form)}
\]

\[
E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{4}{3} & 1 \end{bmatrix}
\]

so we get \( E_2 E_1 A = U \). Then

\[
A = (E_2 E_1)^{-1} U = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{4}{3} & 1 \end{bmatrix} U = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 2 \\ 0 & 3 & -4 \end{bmatrix}
\]

In this example, \( E_1^{-1} E_2^{-1} \) turns out to be unit lower triangular, so we’re done: let \( L = E_1^{-1} E_2^{-1} \), and \( A = LU \) is the decomposition we want.

*Is it just luck that \( E_1^{-1} E_2^{-1} \) turned out to be unit lower triangular? No, it’s because of the particular steps we used in the row reduction: see the discussion below.*
**Example 1 continued:** same coefficient matrix $A$ but different $b$

Now that $A = LU$ is factored, we can solve another equation $Ax = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$ easily:

See above: here only $b$ has changed!

$$
\begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
0 & -\frac{1}{3} & 1 \\
1 & -1 & 2 \\
0 & 3 & -4 \\
0 & 0 & \frac{22}{3}
\end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}
$$

By forward substitution

$$
\begin{align*}
y_1 &= 1 \\
y_2 &= 2 - 2(1) = 0 \\
y_3 &= -\frac{1}{3}(0) = 0
\end{align*}
$$

so $y = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$.

Then the second equation becomes

$$
\begin{bmatrix}
1 & -1 & 2 \\
0 & 3 & -4 \\
0 & 0 & \frac{22}{3}
\end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
$$

and, by back substitution we get

$$
\begin{align*}
x_3 &= 0 \\
3x_2 &= 0 + 4(0) \text{ so } x_2 = 0 \text{ so } x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.
\end{align*}
$$

**When does the method used in Example 2 produce an $LU$ decomposition?**

We can always perform the steps illustrated in Example 2 to get a factorization $A = Z \ast U$, where $U$ is an echelon form of $A$ and $Z$ is a product of elementary matrices (in Example 2, $Z = E_1^{-1} E_2^{-1}$). But $Z$ won’t always turn out to be lower triangular (so that $Z$ might not deserve the name “$L$”). When does the method work?

Suppose (as in Example 2) that the only type of ERO used to row reduce $A$ is

“add a multiple of a row to a lower row.”

Let these EROs correspond to the elementary matrices $E_1, \ldots, E_p$. Then $E_p \cdot \ldots \cdot E_1 A = U$ and therefore $A = E_1^{-1} \cdot \ldots \cdot E_p^{-1} U$.

Each matrix $E_i$ and its inverse $E_i^{-1}$ will be unit lower triangular (why?). Then $L = E_1^{-1} \cdot \ldots \cdot E_p^{-1}$ is also lower triangular with 1’s on its diagonal. (Convince yourself that a product of unit lower triangular matrices is a unit lower triangular matrix.)

Note that i) if a row rescaling ERO had been used, then the corresponding elementary matrix would not have only 1’s on the diagonal — and therefore the product $E_1^{-1} \cdot \ldots \cdot E_p^{-1}$ might not be unit lower triangular.
ii) if a row interchange (“swap”) ERO had been used, then the corresponding elementary matrix would not be lower triangular — and therefore the product $E_1^{-1} \cdot \ldots \cdot E_p^{-1}$ might not be lower triangular.

To summarize: the method illustrated in Example 2 always produces an $LU$ decomposition for $A$ if

the only EROs used to row reduce $A$ to $U$ are of the form

“add a multiple of a row to a lower row.”

The restriction in (**) sounds very restrictive until to stop to think about it:

- When you do row reduction systematically, according to the standard steps as outlined in the text, you never add a multiple of a row to a higher row. The reason for using a row replacement ERO is to create a pivot. In other words, adding multiples of rows only to lower rows doesn’t actually constrain you at all: it’s just “standard procedure” in row reductions.

- “No row rescaling” can be an arithmetic inconvenience, but nothing more than an inconvenience. When row reducing by hand, rescaling a row may be helpful to simplify arithmetic, but rescaling is never necessary to get to an echelon form of $A$. (Of course, row rescaling might be needed to proceed on to the row reduced echelon form — because rescaling may be necessary to get 1’s in the pivot positions.)

**Side comment:** If we did allow row rescaling, it would only mean that we could might end up with a lower triangular “$L$” in which some diagonal entries are not 1’s. From the point of view of solving $Ax = b$, this would not be a disaster. However, the text follows the rule that $L$ should have only 1’s on its diagonal, so we’ll stick to that convention.

But (**) really does impose some restriction on us:: for example, $A = \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix}$ simply cannot be reduced to echelon form using (**) — a row interchange is necessary. Here the method in Example 1 simply will not work to produce an $LU$ decomposition. However (see the last section of the online version of these notes), there is a work-around that is “almost as good” as an $LU$ decomposition in such situations.
To find $L$ more efficiently

If $A$ is large, then row reducing $A$ to echelon form might involve a very large number of elementary matrices $E_i$ in the formula (***)—so that using this formula to find $L$ is too much work. Fortunately, there is a technique, illustrated in Example 3, to write down $L$, entry by entry, as you row reduce $A$. (The text gives a slightly different presentation, and offers an explanation of why it works. It’s really nothing more than careful bookkeeping. Since these notes are just a survey about LU decompositions, we will only write down “the method.”)

Assume $A$ is $m \times n$ and that it can be row-reduced to $U$ following the rule (**). Then $L$ will be a square $m \times m$ unit lower triangular matrix. You can write down $L$ using these steps:

- Write down an “empty” $m \times m$ matrix $L$
- Put 1's on the diagonal of $L$ and 0's in each position above the diagonal
- Whenever you perform a row operation
  
  “add $c$ times row $i$ to a lower row $j$”,

then put $-c$ in the $(j, i)$ position of $L$.

Careful: notice the sign change on $c$, and put $-c$ in the $(j, i)$ position of $L$, not in the $(i, j)$ position

- When $A$ has been reduced to echelon form, stop. If there are any remaining empty positions in $L$, fill them with 0's.

If we apply these steps while row reducing in Example 2, here's what we'd get:

- $A$ was $3 \times 3$, so start with $L = \begin{bmatrix} 1 & 0 & 0 \\ \ast & 1 & 0 \\ \ast & \ast & 1 \end{bmatrix}$
- The row operations we used in Example 1 were:
  
  add $-2$ (row 1) to row 2, so put 2 in the $(2, 1)$ position in $L$: that is, $L_{21} = 2$
  add $\frac{-4}{3}$ (row 2) to row 3, so put $\frac{4}{3}$ in the $(3, 2)$ entry in $L$: that is, $L_{32} = \frac{4}{3}$

Now we have $L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ \ast & \frac{4}{3} & 1 \end{bmatrix}$

- Fill in the rest of $L$ with 0's, to get

  $L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & \frac{4}{3} & 1 \end{bmatrix}$, the same as we got by multiplying elementary matrices
Example 3 In this example, $A$ is a little larger and not square. But otherwise, the calculations are parallel to the ones we did earlier. If possible,

find an $LU$ factorization of $A = \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 1 \\ -2 & 0 & -1 & -2 & 8 & 3 \\ -1 & 0 & -1 & -1 & 3 & 1 \end{bmatrix}$ and use it to solve

\[ Ax = b = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix} \]

\[ A = \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 1 \\ -2 & 0 & -1 & -2 & 8 & 3 \\ -1 & 0 & -1 & -1 & 3 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 1 \\ 0 & 0 & 5 & -2 & 12 & 5 \\ -1 & 0 & -1 & -1 & 3 & 1 \end{bmatrix} \]

\[ \sim \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 1 \\ 0 & 0 & 5 & -2 & 12 & 5 \\ 0 & 0 & 0 & & -\frac{2}{5} & \frac{1}{5} & 0 \end{bmatrix} = U \]

$U$ is in echelon form, and we used only EROs of type (**)..

To find $L$, which must be $3 \times 3$: start with $L = \begin{bmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{bmatrix}$. The row operations we used were

\[
\begin{aligned}
&\text{add 2* (row 1) to (row 2)} \\
&\text{add 1* (row 1) to (row 3)} \\
&\text{add } -\frac{7}{5} \text{ (row 2) to (row 3)}
\end{aligned}
\]

so $L = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & \frac{2}{5} & 1 \end{bmatrix}$. Then

\[ A = \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 1 \\ -2 & 0 & -1 & -2 & 8 & 3 \\ -1 & 0 & -1 & -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} L & U \end{bmatrix} \]

\[ \begin{bmatrix} L & U \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 3 & 0 & 2 & 1 \\ -2 & 1 & 0 & 0 & 0 & 5 & -2 & 12 & 5 \\ -1 & \frac{2}{5} & 1 & 0 & 0 & 0 & -\frac{1}{5} & \frac{1}{5} & 0 \end{bmatrix} \]

We can solve $Ax = b = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix}$ by writing
\[ \begin{aligned}
L y &= b, \quad \text{that is} \\
\begin{bmatrix}
1 & 0 & 0 \\
-2 & 1 & 0 \\
-1 & \frac{2}{5} & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3
\end{bmatrix}
= b =
\begin{bmatrix}
-1 \\
2 \\
3
\end{bmatrix}
\end{aligned} \]

\[ y = U x, \quad \text{that is} \]
\[ \begin{bmatrix}
1 & 0 & 3 & 0 & 2 & 1 \\
0 & 0 & 5 & -2 & 12 & 5 \\
0 & 0 & 0 & -\frac{1}{5} & \frac{1}{5} & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{bmatrix}
= \begin{bmatrix}
y_1 \\
y_2 \\
y_3
\end{bmatrix} \]

The first equation gives
\[ y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 + 2(y_1) \\ 3 + y_1 - \frac{2}{3}y_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}, \text{ and then the second} \]

\[ \begin{aligned}
\begin{bmatrix}
1 & 0 & 3 & 0 & 2 & 1 \\
0 & 0 & 5 & -2 & 12 & 5 \\
0 & 0 & 0 & -\frac{1}{5} & \frac{1}{5} & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{bmatrix}
= \begin{bmatrix}
y_1 \\
y_2 \\
y_3
\end{bmatrix}. \text{ So the solution is} \]

\[ x = \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{bmatrix} = \begin{bmatrix}
11 + 4x_5 + 2x_6 \\
x_2 \\
-2x_5 - x_6 - 4 \\
x_4 \\
x_5 \\
x_6
\end{bmatrix} = \begin{bmatrix}
11 \\
0 \\
-4 \\
-10 \\
0 \\
0
\end{bmatrix} + x_2 \begin{bmatrix}
0 \\
1 \\
0 \\
0 \\
0 \\
0
\end{bmatrix} + x_4 \begin{bmatrix}
4 \\
0 \\
-2 \\
-10 \\
0 \\
0
\end{bmatrix} + x_5 \begin{bmatrix}
0 \\
0 \\
-2 \\
0 \\
0 \\
0
\end{bmatrix} + x_6 \begin{bmatrix}
2 \\
0 \\
-1 \\
0 \\
0 \\
0
\end{bmatrix}. \]

\textbf{MATERIAL BEYOND THIS POINT IN THESE NOTES IS OPTIONAL.}
A Useful Related Decomposition

Sometimes row interchanges are simply unavoidable to reduce $A$ to echelon form. Other times, even when row interchanges could be avoided, they are desirable in computer computations to reduce roundoff errors (see the Numerical Note about partial pivoting in the textbook, p. 17).

What happens if row interchanges are used? We can still get a factorization that resembles $A = LU$ and every bit as useful for solving equations. Here's how.

*Note: as before, all row replacement operations follow the rule (**), and we do not allow rescaling of rows.*

1) Look ahead (on scratch paper!) at what you need to do to reduce $A$ to an echelon form, using (***) and, if necessary, row interchanges. Make note what row interchanges will be used. Then go back to the beginning and “rearrange rows” in $A$ — that is, do all these row interchanges first. The result is “$A$ rearranged.” Let’s look at this in more detail:

Suppose the row interchanges we use are represented by the elementary matrices $E_1, ..., E_k$. Then the “rearranged $A$” is just $E_k \cdot ... \cdot E_1 A = PA$, where

$P = E_k \cdot ... \cdot E_1$.

Since $P = E_k \cdot ... \cdot E_1 = E_k \cdot ... \cdot E_1 \cdot I$ we see that $P$ is just the final result of when the same row interchanges are performed on $I$ — that is, $P$ is just “$I$ with some rows rearranged.” $P$ is called a permutation matrix; the multiplication $PA$ rearranges (“permutes”) the rows of $A$ in exactly the same way that the rows of $I$ were permuted to create $P$. A permutation matrix $P$ is the same as “a square matrix with exactly one 1 in each row and exactly one 1 in each column.” Why?

$P$ is a product of invertible (elementary) matrices, so $P$ is invertible and $P^{-1} = E_1^{-1} \cdot ... \cdot E_p^{-1}$ is also a permutation matrix (why? what does each $E_i^{-1}$ do to $I$?). (If you know the steps to write down $P$, it’s just as easy to actually write down the matrix $P^{-1}$. Or, perhaps you can convince yourself that $P^{-1} = P^T$.)

Since $P^{-1} PA = A$, multiplication by $P^{-1}$ rearranges the rows of $PA$ to restore the original matrix $A$.

2) The matrix “$A$ rearranged” is $PA$, and $PA$ can be row reduced to an echelon form only using row replacements (***) — no need for further row interchanges. Therefore we can use our previous method to find an $LU$ decomposition of $PA$:

$$PA = LU$$

3) Then $A = (P^{-1} L) U$. Here, $P^{-1} L$ is a permuted lower triangular matrix, meaning “lower triangular with some rows interchanged.” MATLAB Help gives $P^{-1} L$ the cute name “psychologically lower triangular matrix,” a term never used anywhere else to my knowledge.

This factorization is just as useful for equation solving as the $LU$ decomposition. For example, we can solve $Ax = (P^{-1} L) U x = b$ by first substituting $y = U x$. It’s still easy to solve $(P^{-1} L)y = b$ because $P^{-1} L$ is psychologically lower triangular (see illustration below), and
then knowing $y$, it's easy to solve $U\mathbf{x} = y$ because $U$ is in echelon form.

For example, suppose $(P^{-1}L)y = b$ is the equation

$$
\begin{bmatrix}
2 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
2 & 2 & 1 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
1 \\
1 \\
0
\end{bmatrix}
$$

permuted lower triangular matrix

We can simply imitate forward substitution, but solve working the rows in the order that (rearranged) would make a lower triangular matrix (simplest row first, the row that involves only the variable $y_1$). That is,

$$
\begin{aligned}
y_1 &= 1, & \text{from the second row, then} \\
y_2 &= 0 - 2y_1 = -2, & \text{from the first row, then} \\
y_3 &= 0 - y_1 - y_2 = 1, & \text{from the fourth row, then} \\
y_4 &= 1 - 2y_1 - 2y_2 - y_3 = 2, & \text{from the third row}
\end{aligned}
$$

(In some books or applications, when you see $A = LU$ it might even be assumed that $L$ is permuted lower triangular rather than lower triangular. Be careful when reading other books to check what the author means.)

OR you could solve $A\mathbf{x} = (P^{-1}L)U\mathbf{x} = b$ as follows:

The equation $PA\mathbf{x} = Pb$ has exactly the same solutions as $A\mathbf{x} = b$ because:

- If $PA\mathbf{x} = Pb$ is true, then multiplying both sides by $P^{-1}$ shows that $\mathbf{x}$ is also a solution for $A\mathbf{x} = b$, and
- If $A\mathbf{x} = b$ is true, then multiplying by $P$ shows that $PA\mathbf{x} = Pb$ is also true.

If you think in terms of writing out the linear systems of equations: $PA\mathbf{x} = Pb$ is the same system of equations as $A\mathbf{x} = b$ but with the equations listed in some other interchanged order determined by the permutation matrix $P$.

Since we have an $LU$ decomposition for $PA$, solving $PA\mathbf{x} = LU\mathbf{x} = Pb$ for $\mathbf{x}$ is easy.

**Example 3** Let $A = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 1 & 1 \\ 1 & 4 & 4 \\ 0 & 1 & 1 \end{bmatrix}$.

1) On scratch paper, do enough steps row reducing $A$ to echelon form (no row rescaling allowed, and, following standard procedure, only adding multiples of rows to lower rows) to see what row interchanges, if any, are necessary. It turns out that only one, interchanging rows 2 and 4 at a certain stage, will be enough.
2) So go back to the start and perform that row interchange first, creating

\[ \begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & 1 \\ 1 & 4 & 4 \\ 1 & 1 & 1 \end{bmatrix}, \text{ where } P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \]

3) Row reduce \( PA \) to an echelon form \( U \), keeping track of the EROs used:

\[
\begin{bmatrix}
1 & 1 & 2 \\
0 & 1 & 1 \\
0 & 3 & 2 \\
0 & 0 & 1
\end{bmatrix}
\xrightarrow{(1)}
\begin{bmatrix}
1 & 1 & 2 \\
0 & 1 & 1 \\
0 & 3 & 2 \\
0 & 0 & 1
\end{bmatrix}
\xrightarrow{(2)}
\begin{bmatrix}
1 & 1 & 2 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}
\xrightarrow{(3)}
\begin{bmatrix}
1 & 1 & 2 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
\frac{1}{2}
\begin{bmatrix}
1 & 1 & 2 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}
\]

The row operations were:

1) add \(-1\times\text{row}(1)\) to \text{row} 3
2) add \(-1\times\text{row}(1)\) to \text{row} 4
3) add \(-3\times\text{row}(2)\) to \text{row} 3
4) add \(-1\times\text{row}(3)\) to \text{row} 4

Using the method described earlier “to find \( L \) more efficiently” we can write down \( L \) as we row reduce \( PA \):

\[
\begin{bmatrix}
1 & 0 & 0 \\
* & 1 & 0 \\
* & * & 1 \\
* & * & 1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
1 & 0 & 0 \\
* & 1 & 0 \\
* & 1 & 0 \\
* & * & 1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
1 & 0 & 0 \\
* & 1 & 0 \\
1 & 1 & 0 \\
1 & * & 1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1
\end{bmatrix}
\]

Then

\[
\begin{bmatrix}
1 & 1 & 2 \\
0 & 1 & 1 \\
1 & 4 & 4 \\
1 & 1 & 1
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
1 & 3 & 1 \\
1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 2 \\
0 & 1 & 1 \\
0 & 0 & -1 \\
0 & 0 & 0
\end{bmatrix}
= L \rightarrow U
\]
To solve $Ax = \begin{bmatrix} 2 \\ 1 \\ 3 \\ 7 \end{bmatrix}$, write $PAx = LUx = Pb = \begin{bmatrix} 2 \\ 7 \\ 3 \\ 1 \end{bmatrix}$ (rows 2,4 of $b$ interchanged, as for $A$ earlier). We then solve $LUx = \begin{bmatrix} 2 \\ 7 \\ 3 \\ 1 \end{bmatrix}$ in the same way as before.

**Factoring $A$ with slight variations from $A = LU$**

**LDU decompositions** Suppose we can row reduce $A$ using only EROs described in (***) and get the factorization $A = LU$. If we loosen up and then allow rescaling of the rows of $U$, we can convert all the leading entries (pivot positions) in $U$ to 1's. This amounts to a factorization $A = LDU$ where, as before, $L$ is unit lower triangular, $D$ is a (square) diagonal matrix and $U$ is an echelon form of $A$ which (by rescaling) has only 1’s in its pivot positions. Since $L$ is unit lower triangular, this produces more symmetry in appearance between $L$ and $U$. (For example: suppose $A$ is square and invertible, and that $A = LDU$ as described. Then $L$ and $U$ will both be square, with $L$ unit lower triangular and $U$ unit upper triangular — 0's below the diagonal and only 1’s on the diagonal. Why must $U$ have that form?) For $U$, this is an aesthetically nicer form, but frankly, it doesn't help much in solving a system of equations.

To illustrate the (easy) details: first factor as $A = LU$. Create a diagonal matrix $D$ as follows:

If Row$_i$ of $U$ has a pivot position:

Let $p_i (\neq 0)$ be the number in the pivot position. Factor $p_i$ out of Row$_i$ of $U$ (leaving a 1 in the pivot position) and use the number $p_i$ as the $i^{th}$ diagonal entry in $D$.

Otherwise, make 0 the $i^{th}$ diagonal element of $D$.

After factoring out the $p_i$'s from the rows of $U$, what's left is a new matrix $U'$ that has only 1's in its pivot positions, and $DU' = U$ (because multiplication on the left by a diagonal matrix just multiplies each row of $U'$ by the corresponding diagonal element of $D$). Therefore $A = LDU'$. 

At this point, we abuse the notation and write $U$ in place of $U'$ — we do this only for neatness, since the rightmost matrix in this factorization is traditionally called $U$: just remember that the $U$ in $A = LDU$ usually isn't the same as the $U$ in the original factorization $A = LU$.

$A = LDU$ is cleverly called an *LDU* decomposition of $A$. Here's an illustration using the matrices from Example 2.

$$ A = \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 1 \\ -2 & 0 & -1 & -2 & 8 & 3 \\ -1 & 0 & -1 & -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 2/5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 1 \\ 0 & 0 & 5 & -2 & 12 & 5 \\ 0 & 0 & 0 & -1/5 & 1/5 & 0 \end{bmatrix} $$
Note for those who use MATLAB

MATLAB has a command \( [L, U, P] = \text{lu}(A) \) (\( \text{lu} \) is lowercase in MATLAB)

Unless you're very comfortable with MATLAB and the material, you should probably avoid confusion by only using the Matlab command on square matrices \( A \) (See the technical note, below.)

In that case, MATLAB should give you

1) a square unit lower triangular \( L \)
2) \( U \) in echelon form, and
3) a permutation matrix \( P \)

for which \( PA = LU \).

If it's possible simply to factor \( A = LU \), then MATLAB might just give \( P = I = \) the identity matrix. However, sometimes MATLAB will permute some rows even when it's not mathematically necessary (because doing so, in large complicated problems, will help reduce roundoff errors). See the Numerical Note about partial pivoting in the textbook, p. 17).

If \( A \) is not square, then MATLAB gives a result so that \( PA = LU \) is still true, but the shapes of the matrices may vary from our discussion.

Technical note from a department MATLAB expert:

If \( A \) is \( m \times n \), then \( [L, U, P] = \text{lu}(A) \) should return an \( m \times m \) unit lower triangular \( L \), \( m \times n \) upper triangular \( U \), and \( m \times m \) permutation \( P \) satisfying \( PA=LU \).

In fact, the dimensions of what MATLAB returns depends on whether \( m = n \), \( m < n \), or \( m > n \). Evidently the algorithm performs Gaussian elimination with partial pivoting as long as it can on \( A \), in place, then extracts relevant parts of the changed \( A \) as \( L \) and \( U \), with \( 1 \)'s inserted into \( L \). \( P \) is stored as a vector but output as a matrix. Thus \( U \) will come out as \( m \times n \) and \( P \) will be \( m \times m \), but \( L \) will be \( m \times k \) with \( k = \min(m, n) \) because its zero-rows are superfluous. V. Wickerhauser)
Here's a small-scale illustration of another way in which the $LU$ decomposition can be helpful in some large, applied computations. A band matrix refers to a square $n \times n$ matrix in which all the nonzero entries lie in a “band” that surrounds the main diagonal.

For example, in the following matrix all nonzero entries are in the “band” that consists of one “superdiagonal” above the main diagonal and two “subdiagonals” below the main diagonal. The number of superdiagonals in the band is called the upper bandwidth (1, in the example below) and the number of subdiagonals in the band is the lower bandwidth (2, in the example). The total number of diagonals in the band is called the bandwidth (4, in the example).

\[
\begin{bmatrix}
1 & 3 & 0 & 0 \\
1 & 0 & 2 & 0 \\
1 & 4 & 3 & 0 \\
2 & 1 & 1 & 1 \\
& 1 & 2 & 4
\end{bmatrix}
\] (all other entries = 0)

A diagonal matrix such as
\[
\begin{bmatrix}
3 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 6
\end{bmatrix}
\]
is an extreme example for which bandwidth = 1.

Use of the term “band matrix” is a bit subjective: in fact, we could think of any matrix as a band matrix — if, say, all entries are nonzero, then the matrix is just one that has the largest possible bandwidth. In practice, however, the term “band matrix” is used to refer to a matrix in which the bandwidth is relatively small compared to the size of the matrix. Therefore the matrix is “sparse” — a relatively large number of entries are 0.

Look at the simple problem about steady heat flow in Exercises 33-4 in Section 1.1 in the text; a more larger problem, set up the same way, then occurs in Exercise 31, Section 2.5.
Given the temperatures at the nodes around the edge of the plate, we want to approximate the temperature $x_1, \ldots, x_8$ at the interior nodes numbered 1,\ldots,8. This leads, as in Exercise 33, Section 1.1 to an equation $Ax = b$ with a band matrix $A$:

$$
\begin{bmatrix}
4 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & 0 & -1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 4 & -1 & -1 & 0 & 0 & 0 \\
0 & -1 & -1 & 4 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 & 4 & -1 & -1 & 0 \\
0 & 0 & 0 & -1 & -1 & 4 & 0 & -1 \\
0 & 0 & 0 & 0 & -1 & 0 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & 4
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8
\end{bmatrix}
= 
\begin{bmatrix}
5 \\
15 \\
0 \\
10 \\
0 \\
10 \\
20 \\
30
\end{bmatrix}
$$

If we can find an $LU$ decomposition of $A$, then $L$ and $U$ will also be band matrices (Why? Think about how you find $L$ and $U$ from $A$). Using the MATLAB command $[L, U, P] = lu(A)$ gives an $LU$ decomposition. Rounded to 4 decimal places, we get

$L = $

$$
\begin{bmatrix}
1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.2500 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.2500 & -0.0667 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.2667 & -0.2857 & 1.0000 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.2679 & -0.0833 & 1.0000 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.2917 & -0.2921 & 1.0000 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.2697 & -0.0861 & 1.0000 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.2948 & -0.2931 & 1.0000
\end{bmatrix}
$$
The decomposition can be used to solve \( Ax = b \), as we showed earlier in these notes. The important thing in the example is not actually finding the solution, which turns out to be:

\[
U = 
\begin{bmatrix}
4.0000 & -1.0000 & -1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.7500 & -0.2500 & -1.0000 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3.7333 & -1.0667 & -1.0000 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3.4286 & -0.2857 & -1.0000 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 3.7083 & -1.0833 & -1.0000 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3.3919 & -0.2921 & -1.0000 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 3.7052 & -1.0861 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.3868 & 0
\end{bmatrix}
\]

The \( LU \) decomposition can be used to solve \( Ax = b \), as we showed earlier in these notes. The important thing in the example is not actually finding the solution, which turns out to be:

\[
\begin{bmatrix}
3.9569 \\
6.5885 \\
4.2392 \\
7.3971 \\
5.6029 \\
8.7608 \\
9.4115 \\
12.0431
\end{bmatrix}
\]

What's interesting and useful to notice:

- From earlier in these notes: if we need to solve \( Ax = b \) many times for different vectors \( b \), then factoring \( A = LU \) saves time.

- A new observation is that for a very large \( n \times n \) “sparse” band matrix \( A \) (perhaps thousands of rows and columns but with a relatively narrow band of nonzero entries), we can store \( A \) with much less computer space than is needed for a general \( n \times n \) matrix: we only need to store the entries in the upper and lower bands, and two numbers for the upper and lower bandwidths; no need to waste space storage space for all the other matrix entries which are known to be 0’s. Since \( L \) and \( U \) are also sparse band matrices, they are also relatively economical to store.

- Using \( A, L \) and \( U \) can let us avoid computations that use \( A^{-1} \). Even if \( A \) is a sparse band matrix, \( A^{-1} \) may have no band structure and take up a lot more storage space than \( A, L, \) and \( U \). For instance, in the preceding example,
\[
A^{-1} = \begin{bmatrix}
0.2953 & 0.0866 & 0.0945 & 0.0509 & 0.0318 & 0.0227 & 0.0100 & 0.0082 \\
0.0866 & 0.2953 & 0.0509 & 0.0945 & 0.0227 & 0.0318 & 0.0082 & 0.0100 \\
0.0945 & 0.0509 & 0.3271 & 0.1093 & 0.1045 & 0.0591 & 0.0318 & 0.0227 \\
0.0509 & 0.0945 & 0.1093 & 0.3271 & 0.0591 & 0.1045 & 0.0227 & 0.0318 \\
0.0318 & 0.0227 & 0.1045 & 0.0591 & 0.3271 & 0.1093 & 0.0945 & 0.0509 \\
0.0227 & 0.0318 & 0.0591 & 0.1045 & 0.1093 & 0.3271 & 0.0509 & 0.0945 \\
0.0100 & 0.0082 & 0.0318 & 0.0227 & 0.0945 & 0.0509 & 0.2953 & 0.0866 \\
0.0082 & 0.0100 & 0.0227 & 0.0318 & 0.0509 & 0.0945 & 0.0866 & 0.2953 \\
\end{bmatrix}
\]