Math 110 - Fall 05 - Lectures notes # 20 - Oct 14 (Friday)

We will cover the contents of Chapter 3 slightly differently, than the textbook, showing how to compute

- range(A), rank(A), nullspace(A), nullity(A),
- $A^{-1}$ (if it exists)
- solve $Ax=b$ (determine if there are no solutions, 1 solution, or infinitely many solutions).

all by computing a single "matrix factorization" of $A$, called the "LU factorization" or "LU decomposition", which is a formula for $A$ as product of much simpler matrices.

The algorithm for computing this factorization is just Gaussian elimination, where we keep track of all the elementary matrix operations we do:

- reordering rows and/or columns
- adding a multiply of one row to another
- multiplying a row or column by a scalar

as a product of simple matrices. When you use computer software to do matrix operations, it will return your answer as a product of simpler matrices much like the ones we show here.

LU decomposition plays an important role in government high technology policy: the largest computers in the world (called "supercomputers") are regularly measured by how fast they can solve $Ax=b$ using LU decomposition on very large matrices, with the ratings appearing on the web site www.top500.org. This web site keeps track of the 500 fastest computers anywhere in the world (at least those not at secret government sites).

For example, as of June 2005, the top computer was an IBM Blue Gene/L, consisting of 65,536 parallel processors. The machine solved a linear system of $n = 1,277,951$ equations using the LU decomposition algorithm described below at a speed of 136.8 Teraflops, that is 136.8 trillion = 136.8 x 10^12 arithmetic operations per second. Since the number of arithmetic operations required by the algorithm will turn out to be about $2/3 n^3$, the time required to solve this single linear system was

$$(2/3)*(1,277,951)^3 / 136.8x10^{12} = 10161 \text{ seconds} \sim 2.8 \text{ hours}$$

Recall that multiplying 2 matrices takes about $2n^3$ operations, 3 times more operations.

A few years ago the fastest machine on the Top 500 list was the NEC "Earth Simulator", which ran at a speed of 35.9 Teraflops, as fast as the next four machines combined. This caught the attention of parts
of the US government responsible for national security, because NEC is a Japanese company, and one national security goal of the US government is to have better technology, including faster computers, than anyone else in the world. This is because these computers are used for breaking secret codes, designing weapons, and other national security purposes. Indeed, the largest computers in the US have been built for these purposes for decades. The speed of the NEC machine led to a flurry of government studies and reports to analyze the situation, for example www.sc.doe.gov/ascr.FOSCfinalreport.pdf. More aspects of implementing linear algebra algorithms on parallel machines are discussed in the course CS267, see www.cs.berkeley.edu/~demmel/cs267_Spr05.

One of the most widely used software packages for solving linear algebra problems is called LAPACK (its parallel version is ScaLAPACK); for example it is used within Matlab. This software was produced in a joint project between UC Berkeley, the University of Tennessee, and other groups. We have recently been funded to produce a new version. See www.netlib.org/lapack for the current software.

Ex: Suppose $L$ is square and lower triangular:

$$L = \begin{bmatrix}
1 & 0 & 0 \\
1 & 2 & 0 \\
3 & -1 & 3
\end{bmatrix}$$

Then we can solve $Lx = b$ for $x$ given any $b$ by "substitution"

- Solve $1*x_1 = b_1$ for $x_1$
- Solve $1*x_1 + 2*x_2 = b_2$ for $x_2$
- Solve $3*x_1 - 1*x_2 + 3*x_3 = b_3$ for $x_3$

As long as each $L_{ii}$ is nonzero, so we can divide by it, we can solve any $Lx=b$ for $x$ and get a unique answer. In particular, the only solution of $Lx = 0$ is $x=0$, so that $L$ is one-to-one. This implies that $L$ is invertible.

We can compute $L^{-1}$ by solving for it one column at a time:

- Let $X = L^{-1}$, so $L*X = I$, and $e_j = j$-th standard basis vector, so $L*X*e_j = I*e_j$, or
- $L*(X*e_j) = e_j$ or $L*(\text{column } j \text{ of } X) = e_j$

so we can solve for $X = L^{-1}$ one column at a time.

Ex: Suppose $U$ is square, upper triangular, and each $U_{ii}$ is nonzero.

Then we can similarly solve any $U*x=b$ for a unique $x$ given $b$ by substitution, solving first for $x_n$, then $x_{n-1}$, and so on. We can also compute $U^{-1}$ as above.
Ex: Suppose $A = L \cdot U$, where $L$ and $U$ are $n$ by $n$ and nonsingular. Then $A$ is invertible (why?), and

$$A^{-1} = (L \cdot U)^{-1} = U^{-1} \cdot L^{-1}$$

and solving $A \cdot x = b$ for $x$ given $b$ means computing

$$x = A^{-1}b$$

$$= (U^{-1} \cdot L^{-1})b$$

$$= U^{-1} \cdot (L^{-1}b) \quad \text{... by associativity}$$

$$= U^{-1} \cdot y$$

... where we compute $y = L^{-1}b$ i.e. solve $L \cdot y = b$ by substitution where we finally compute $x = U^{-1}y$ i.e. solve $U \cdot x = y$ by substitution again. In other words, if we know $A = L \cdot U$ where $L$ and $U$ are invertible triangular matrices, we can solve $A \cdot x = b$ by 2 steps:

1. solve $L \cdot y = b$ by substitution
2. solve $U \cdot x = y$ by substitution

If we want to compute $A^{-1}$ we use the same idea as above, solving $A \cdot X = I$ for $X$ one column of $X$ at a time.

Stop&Ask: Suppose $A = U \cdot L$ where $U$ and $L$ are $n$ by $n$ and invertible.

How do we solve $A \cdot x = b$?

It may seem strange to expect to know $A$ in the form $A = L \cdot U$, but this is in fact exactly what Gaussian elimination applied to $A$ computes.

Ex: $A = \begin{bmatrix} 3 & -1 & 2 \\ 6 & 0 & 10 \\ -3 & -3 & -17 \end{bmatrix}$

To apply Gaussian elimination, we do the following:

for $i = 1$ to $n-1$ (3-1 = 2 in this case)
subtract multiples of row $i$ from rows $i+1$ through $n$,
in order to zero out $A$ below the diagonal

Subtract 2 \ * \ \text{row 1} \ \text{from row 2}
Subtract -1 \ * \ \text{row 1} \ \text{from row 3},
yielding $A' = \begin{bmatrix} 3 & -1 & 2 \\ 0 & 2 & 6 \\ 0 & -4 & -15 \end{bmatrix}$

Subtract -2 \ * \ \text{row 2} \ \text{from row 2},
yielding $A'' = \begin{bmatrix} 3 & -1 & 2 \\ 0 & 2 & 6 \\ 0 & 0 & -3 \end{bmatrix}$

Note that $A'' = U$ is upper triangular. Let's take the multipliers we computed and put them into a lower triangular matrix $L$
Let 
\[
L = \begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
-1 & -2 & 1
\end{bmatrix}
\]

Then we can confirm that \(L \cdot U = A\). We will see that this is true in general, as long as we never get a 0 on the diagonal.

ASK & WAIT: What happens if \(A_{11} = 0\)? What happens at any point if some diagonal entry = 0?

If \(A\) is \(m\)-by-\(n\), we will only get a zero on the diagonal when \(\text{rank}(A) < \min(m,n)\), as we will see. First we will see what happens assuming this is not the case, and then show the most general case.

We will use induction. To do so, we need some basic facts about "block matrices", which are a useful notation to keep track of what parts of the matrix we're working on:

Def. Let \(A\) by \(m\) by \(n\), where \(m = m_1 + m_2\) and \(n = n_1 + n_2\). Then we can write \(A\) as a "block matrix"
\[
A = \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\]

where \(A_{11}\) and \(A_{21}\) have \(m_1\) columns, \(A_{12}\) and \(A_{22}\) have \(m_2\) columns
\(A_{11}\) and \(A_{12}\) have \(n_1\) rows, and \(A_{21}\) and \(A_{22}\) have \(n_2\) rows.

We sometimes write the dimensions around the matrix to indicate this:
\[
A = \begin{bmatrix}
A_{11} & A_{12} \end{bmatrix} n_1 \\
A_{21} & A_{22} \end{bmatrix} n_2
\]

Here is perhaps the most useful fact about block matrices, that when we multiply them, we can just multiply and add the blocks, as though they were scalars:

Lemma 1: Let \(A\) and \(B\) be block matrices, where \(A\)'s column block dimensions \(m_1\) and \(m_2\) match \(B\)'s row block dimensions:
\[
A = \begin{bmatrix}
A_{11} & A_{12} \end{bmatrix} n_1 \\
A_{21} & A_{22} \end{bmatrix} n_2
\]
\[
B = \begin{bmatrix}
B_{11} & B_{12} \end{bmatrix} m_1 \\
B_{21} & B_{22} \end{bmatrix} m_2
\]

This means that matrix products like \(A_{11} \cdot B_{11}\) are defined, and in fact \(A \cdot B\) can be written as the block matrix
\[
A \cdot B = \begin{bmatrix}
A_{11} \cdot B_{11} + A_{12} \cdot B_{21} & A_{11} \cdot B_{12} + A_{12} \cdot B_{22} \\
A_{21} \cdot B_{11} + A_{22} \cdot B_{21} & A_{21} \cdot B_{12} + A_{22} \cdot B_{22}
\end{bmatrix} n_1 \\
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]
\[
\]
Note that this is the usual formula for multiplying 2 by 2 matrices, when all the block \( A_{ij} \) and \( B_{ij} \) are 1 by 1, i.e. scalars.

Proof: (Picture) When \((i,j)\) is in the top left block, we see
\[
(A*B)_{ij} = \sum_{k=1}^{m} A_{ik}B_{kj}
\]
\[
= \sum_{k=1}^{m1} A_{ik}B_{kj} + \sum_{k=m1+1}^{m} A_{ik}B_{kj}
\]
\[
= \sum_{k=1}^{m1} A_{11}B_{11} + \sum_{k=1}^{m2} A_{12}B_{21}
\]
or the leading \( n1 \) by \( p1 \) block of \( A*B = A_{11}B_{11} + A_{12}B_{21} \)
The other blocks of \( A*B \) are similar.

Corollary 1: Let
\[
L = \begin{bmatrix}
I^{m1} & 0 \\
L_{21} & I^{m2}
\end{bmatrix}
\]
be a lower triangular matrix, where \( I^{m1} \) is an \( m1 \) by \( m1 \) identity matrix, \( I^{m2} \) is an \( m2 \) by \( m2 \) identity matrix, and \( 0 \) is an \( m1 \) by \( m2 \) zero matrix. Then
\[
L^{-1} = \begin{bmatrix}
I^{m1} & 0 \\
-L_{21} & I^{m2}
\end{bmatrix}
\]
Proof: We just confirm \( L*L^{-1} = I^m = m \) by \( m \) identity matrix:
The dimensions of the blocks match, letting us use Lemma 1 to compute
\[
\begin{bmatrix}
I^{m1} & 0 \\
L_{21} & I^{m2}
\end{bmatrix} \begin{bmatrix}
I^{m1} & 0 \\
-L_{21} & I^{m2}
\end{bmatrix}
\]
\[
= \begin{bmatrix}
I^{m1}I^{m1} + 0 & I^{m1}0 + 0 \cdot I^{m2} \\
L_{21}I^{m1} + I^{m2}(-L_{21}) & L_{21}0 + I^{m2}I^{m2}
\end{bmatrix}
\]
\[
= \begin{bmatrix}
I^{m1} + 0 \\
L_{21} - L_{21}
\end{bmatrix}
\]
\[
= \begin{bmatrix}
I^{m1} \\
0
\end{bmatrix}
\]
\[
= I^m
\]

Now we can show how to compute a factorization \( A = LU \) assuming we never get a zero on the diagonal. We present it for \( A \) rectangular.

Thm 1 (Gaussian Elimination, assuming no 0s appear on the diagonal)
Let \( A \) be \( m \) by \( n \). Then there is a lower triangular matrix \( L \) and upper triangular matrix \( U \) such that \( A = L*U \):
(1) When $A$ is square ($m=n$) then $L$ and $U$ are also $n$ by $n$
(2) When $A$ has more rows than columns ($m>n$), then $L$ is $m$ by $n$ and $U$ is $n$ by $n$
(picture)
(3) When $A$ has more columns than rows ($m<n$), then $L$ is $m$ by $m$ and $U$ is $m$ by $n$
(picture)
In fact, we can pick $L$ to have ones on its diagonal.
This is also called "LU factorization".

Proof:
Base cases: When $\min(m,n)=1$, we pick $L$ and $U$ so $A=L*U$ as follows:
(1) When $A = \begin{bmatrix} a_{11} \end{bmatrix}$ is 1 by 1, choose $L = 1$ and $U = A$
(2) When $m > n = 1$, so $A = \begin{bmatrix} a_{11} \end{bmatrix}$ with $a_{11} \neq 0$, pick $L = \begin{bmatrix} 1 \\ a_{21} \\ ... \\ a_{m1} \end{bmatrix}$
and $U = a_{11}$
(3) When $m = 1 < n$, so $A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \end{bmatrix}$ with $a_{11} \neq 0$,
pick $L = 1$ and $U = A$

Now we do induction. We assume the theorem is true for $m-1$ by $n-1$ matrices,
and prove it is true for an $m$ by $n$ matrix $A$. By assumption $a_{11} \neq 0$.
We write $A$ as a block matrix (note that $A_{11}$ is just $a_{11}$):
$$
\begin{bmatrix} 1 & \ldots & \ldots & 1 \\
A_{11} & A_{12} & \ldots & 1 \\
A_{21} & A_{22} & \ldots & \ldots \\
\vdots & \vdots & \ddots & \ddots \\
A_{m1} & \ldots & \ldots & A_{m1} \\
\end{bmatrix}
$$
Subtracting a multiple of row 1 from row $i$ to zero out $a_{i1}$ means the
multiplier must be $a_{i1}/a_{11}$, since this changes the $(i,1)$ entry from $a_{i1}$
to $a_{i1} - (a_{i1}/a_{11})*a_{11} = 0$. We express this by premultiplying $A$ by
$$
L' = \begin{bmatrix} 1 & \ldots & \ldots & 1 \\
-a_{21}/a_{11} & 1 & \ldots & 0 \\
-a_{31}/a_{11} & 0 & 1 & \ldots \\
\vdots & \vdots & \ddots & \ddots \\
-a_{m1}/a_{11} & \ldots & 0 & 1 \\
\end{bmatrix}
$$
where $-A_{21}/A_{11}$ is $m-1$ by 1, and $0$ is 1 by $m-1$. Lemma 1 lets us multiply
$$
L' * A = \begin{bmatrix} 1 & \ldots & \ldots & 1 \\
-a_{21}/A_{11} & 1 & \ldots & 0 \\
-a_{31}/A_{11} & 0 & 1 & \ldots \\
\vdots & \vdots & \ddots & \ddots \\
-a_{m1}/A_{11} & \ldots & 0 & 1 \\
\end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\
A_{21} & A_{22} \\
\vdots & \vdots \\
A_{m1} & \ldots \ldots \ldots \\
\end{bmatrix}
$$
confirming we have zeroed out the first column beneath the diagonal as desired.

Another way to see what it means to multiply \( L' \cdot A \) is to look at it row by row:
row \( j \) of \( L' \cdot A = e_j^t \cdot (L' \cdot A) \) ...
= \( e_j^t \cdot L' \cdot A \) ...
by associativity
= \( (\text{row } j \text{ of } L') \cdot A \)

When \( j=1 \), row \( j \) of \( L' = e_1^t \) so row 1 of \( L' \cdot A = \text{row 1 of } A \)
When \( j=2 \), row \( j \) of \( L' = [-a_{21}/a_{11}, 1, 0, \ldots, 0] \)
so row 2 of \( L' \cdot A = (-a_{21}/a_{11}) \cdot (\text{row 1 of } A) + (1) \cdot (\text{row 2 of } A) \)

Similarly, for other \( j \), row \( j \) of \( L' \cdot A = (-a_{j1}/a_{11}) \cdot (\text{row 1 of } A) + (\text{row } j \text{ of } A) \)

The quantity \( S = A_{22} - A_{21} \cdot A_{12} / A_{11} \) is called a "Schur complement". Note that \( L' \cdot A \) is closer to upper triangular, because it is zero below the first diagonal. Now \( S \) is \( m-1 \) by \( n-1 \) so we can apply induction to write \( S = L_S \cdot U_S \) (where we again assume no zeros appear on the diagonal):

\[
\begin{bmatrix}
1 & \begin{array}{c}
0 \\
S \\
L_S \\
0
\end{array}
\end{bmatrix}
\begin{bmatrix}
A_{11} & A_{12} \\
0 & S \\
0 & L_S \\
0 & U_S
\end{bmatrix}
\]

Note that the second matrix above is upper triangular; we will call it \( U \). Then \( A = L' \cdot \{ -1 \} \cdot \left[ \begin{array}{cccc}
1 & 0 \\
0 & L_S
\end{array} \right] \cdot U \)

\[
\begin{bmatrix}
1 & 0 \\
0 & L_S
\end{bmatrix}
\begin{bmatrix}
1 & \begin{array}{c}
0 \\
X \cdot \{ n-1 \} \\
0 \cdot L_S
\end{array}
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & L_S
\end{bmatrix}
\]

... where \( X = -A_{21}/A_{11} \) for short

\[
\begin{bmatrix}
1 & \begin{array}{c}
0 \\
X \cdot \{ n-1 \} \\
0 \cdot L_S
\end{array}
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & L_S
\end{bmatrix}
\]

... by Corollary 1

\[
\begin{bmatrix}
1 & \begin{array}{c}
1*1 + 0*0 \\
1*0 + 0*L_S
\end{array}
\end{bmatrix}
\begin{bmatrix}
1 & \begin{array}{c}
1*0 + 0*L_S \\
1*1 + \{ n-1 \}*0
\end{array}
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & \begin{array}{c}
0 \\
-X \cdot L_S
\end{array}
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
-X \cdot L_S
\end{bmatrix}
\]

\[
L \cdot U
\]

where \( L = \begin{bmatrix} 1 & 0 \end{bmatrix} \) is lower triangular with ones on the diagonal as desired.
and with the multipliers from Gaussian elimination below the diagonal. This completes the proof of Thm 1.

Now we generalize this result to the case where zeros may appear on the diagonal. For example, if $A_{11}$ had been 0 above, we would have tried to divide by zero, and it would not have worked. But as long as there is a nonzero somewhere in the matrix, we can always reorder the rows and columns to put that nonzero in the (1,1) entry. If, after the first step, the (2,2) entry is zero, we again permute rows and columns to put a nonzero there if one exists.

We will also use matrix-multiplication, by "permutation matrices", to keep track of this bookkeeping.

Ex: If $A = \begin{bmatrix} 0 & 1 \end{bmatrix}$, then we can't write

\[
\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} = \begin{bmatrix} l_{11}u_{11} & l_{11}u_{12} \\ l_{21}u_{11} & l_{21}u_{12} + l_{22}u_{22} \end{bmatrix}
\]

unless $l_{11} = 0$ or $u_{11} = 0$ to match $a_{11} = 0$.

But this would mean either $a_{12} = 0$ (if $l_{11}=0$) or $a_{21} = 0$ (if $u_{11}=0$), a contradiction. But if we permute $A$'s rows to get $\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$, or $A$'s columns to get $\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$, then we can do LU factorization.

Def: A permutation matrix $P$ is an $n$ by $n$ matrix with exactly one 1 in each row, one 1 in each column, and the other entries equal to 0.

An equivalent definition is that a permutation matrix $P$ is gotten by taking the identity matrix, and permuting its rows (or its columns).

Ex: $P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $P_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $P_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Stop&Ask: How do we construct $P_2$ and $P_3$ by permuting rows or columns of $I$?

Lemma 2: Let $P$ be a permutation matrix. Then

(1) If $y = P*x$, then $y$ has the same entries as $x$ but in a permuted order
(2) If $Y = P*X$, then $Y$ has the same rows as $X$ but in a permuted order
(3) If $Y = X*P$, then $Y$ has the same columns as $X$ but in a permuted order
(4) If $P_1$ and $P_2$ are permutation matrices, so is $P_1*P_2$.
(5) If $P$ is a permutation matrix, so is $P^t = P^{-1}$

Ex: $P_1*[x_1] = [x_1]$, $P_2*[x_1] = [x_2]$, $P_3*[x_1] = [x_3]$

\[
\begin{bmatrix} x_2 \\ x_2 \\ x_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_2 \\ x_2 \\ x_2 \end{bmatrix}
\]

\[
\begin{bmatrix} x_2 \\ x_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 \\ x_2 \end{bmatrix}
\]

\[
\begin{bmatrix} x_2 \\ x_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_1 \end{bmatrix}
\]

8
Proof:

(1) Recall that \( y = Px \) -> \( y_i = e_i^t y = e_i^t (Px) = (e_i^t P) y = \text{(row } i \text{ of } P) y \).

Since row \( i \) of \( P \) is a row of the identity matrix, say \( e_j^t \), then \( y_i = e_j^t x = x_j \). Since each row of \( P \) is a different row of the identity, each \( y_i \) is a different component of \( x \).

(2) Apply (1) to each column of \( Y \) and \( X \), since \( Y = P X \) means that for all \( j \), column \( j \) of \( Y = Ye_j = (Px) e_j = P (X e_j) = P (\text{column } j \text{ of } X) \).

(3), (4) and (5): homework!

Finally, we can do the general case of Gaussian elimination:

Thm 2 (Gaussian Elimination, or LU decomposition in general case)
Let \( A \) be \( m \) by \( n \) and nonzero. Then we can write
\[ A = P_L * L * U * P_R \]
where, for some \( r \) with \( 1 \leq r \leq \min(m,n) \)
- \( P_L \) is an \( m \) by \( m \) permutation matrix (needed to permute \( A \)'s rows)
- \( L \) is an \( m \) by \( r \) lower triangular matrix, (with ones on the diagonal)
- \( U \) is an \( r \) by \( n \) upper triangular matrix (with nonzero diagonal)
- \( P_R \) is an \( n \) by \( n \) permutation matrix (needed to permute \( A \)'s columns)

We can write \( A = P_L * L * U * P_R \) equivalently as \( P_L^t A P_R^t = L * U \), or that \( A \)'s rows and columns can be reordered (by \( P_L^t \) and \( P_R^t \), respectively), so that the reordered matrix has an LU decomposition.

Proof: We use induction as before. This time the base case is more complicated:

Base cases: When \( \min(m,n) = 1 \), we factor \( A \) as follows:

(1) When \( A = [a_{11}] \) is 1 by 1 (and so \( a_{11} \neq 0 \)), pick \( r = 1 \) and write \( A = 1 * 1 * a_{11} * 1 = P_L * L * U * P_R \).

(2) When \( m > n = 1 \), so \( A = [a_{11}] \) with \( a_{11} \neq 0 \), pick \( r = 1 \) and write
\[
[a_{21} \\
[ ... ]
\]
\[ A = I^m \cdot \begin{bmatrix} 1 \\ a_{21} / a_{11} \\ \vdots \\ a_{m1} / a_{11} \end{bmatrix} \cdot a_{11} \cdot 1 \]

\[ = P_L \cdot L \cdot U \cdot P_R \]

When \( m > n = 1 \), with \( a_{11} = 0 \) but some other \( a_{ii} \neq 0 \),
then pick \( r = 1 \) and \( P_L \) to swap entries 1 and i

**ASK & WAIT:** What does \( P_L \) look like?

and write

\[ A = P_L \cdot \begin{bmatrix} 1 \\ a_{21} / a_{ii} \\ a_{31} / a_{ii} \\ \vdots \\ a_{ii} / a_{ii} \end{bmatrix} \cdot 1 \]

\[ = P_L \cdot L \cdot U \cdot P_R \]

(3) When \( m = 1 < n \), so \( A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1m} \end{bmatrix} \) with \( a_{11} \neq 0 \),
then pick \( r = 1 \) and write

\[ A = 1 \cdot 1 \cdot A \cdot I^n \]

\[ = P_L \cdot L \cdot U \cdot P_R \]

When \( m = 1 < n \), so \( A = \begin{bmatrix} a_{11} & a_{21} & \ldots & a_{m1} \end{bmatrix} \)
with \( a_{11} = 0 \) but some other \( a_{ii} \neq 0 \)
then pick \( r = 1 \) and \( P_R \) (as described above) to swap entries 1 and i so

\[ A = 1 \cdot 1 \cdot \begin{bmatrix} a_{ii} & a_{21} & \ldots & a_{11} & \ldots & a_{m1} \end{bmatrix} \cdot P_R \]

\[ = P_L \cdot L \cdot U \cdot P_R \]

Now for the induction. Let \( A \) be \( m \) by \( n \) and assume Theorem 2 is
true for \( (m-1) \) by \( (n-1) \) matrices. If \( A_{11} \neq 0 \),
let \( P'_L = I^m \) and \( P'_R = I^n \). Otherwise, if some other \( A_{ij} \) is nonzero,
pick \( P'_L \) (to swap rows 1 and i) and \( P'_R \) (to swap columns 1 and j)
so that \( P'_L A P'_R \) has a nonzero entry \( (A_{ij}) \) in the \( (1,1) \) position.
Then as above write

\[ P'_L A P'_R = \begin{bmatrix} 1 & \ldots & a_{11} & a_{12} & \ldots & a_{1m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{21} & \ldots & a_{22} & \ldots & \ldots & a_{2m} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{m1} & \ldots & a_{m2} & \ldots & \ldots & a_{mm} \end{bmatrix} \]

\[ = P_L \cdot L \cdot U \cdot P_R \]
\[
= \begin{bmatrix}
1 & 0 \\
A_{21}/A_{11} & I^{\times(n-1)}
\end{bmatrix}
\begin{bmatrix}
A_{11} & A_{12} \\
0 & A_{22} - A_{21}/A_{11}A_{12}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
1 & 0 \\
X & I^{\times(n-1)}
\end{bmatrix}
\begin{bmatrix}
A_{11} & A_{12} \\
0 & S
\end{bmatrix}
\]

where \(X = A_{21}/A_{11}\) and \(S = A_{22} - A_{21}/A_{11}A_{12}\) for short.

There are 2 cases:

(1) If \(S = 0\), then we can confirm that we can write
\[
P'_L A P'_R = \begin{bmatrix}
1 & 0 \\
X & I^{\times(n-1)}
\end{bmatrix}
\begin{bmatrix}
A_{11} & A_{12} \\
0 & 0
\end{bmatrix}
= \begin{bmatrix}
A_{11} & A_{12} \\
X*A_{11} & X*A_{12}
\end{bmatrix}
= \begin{bmatrix} 1 \end{bmatrix}\begin{bmatrix}
A_{11} & A_{12}
\end{bmatrix} = L * U
\]

Thus \(A = (P'_L)^{-1} * L * U * (P'_R)^{-1}\)

\[
P_L = P'_L^t \quad \text{and} \quad P_R = P'_R^t
\]

are permutations by Lemma 2 part (5), is the desired LU factorization.

(2) If \(S\) is nonzero, we can apply the induction hypothesis to \(S\) to get \(S = P_LS * L_S * U_S * P_RS\), where \(L_S (U_S)\) has \(r'\) columns (rows)

Plugging this in we get
\[
A = (P'_L)^t * \begin{bmatrix}
1 & 0 \\
X & I
\end{bmatrix} * \begin{bmatrix}
A_{11} & A_{12} \\
0 & P_LS*L_S*U_S*P_RS
\end{bmatrix}
\]
... substituting for \(S\)

\[
= (P'_L)^t * \begin{bmatrix}
1 & 0 \\
X & I
\end{bmatrix} * \begin{bmatrix}
1 & 0 \\
0 & P_LS*L_S
\end{bmatrix} * \begin{bmatrix}
A_{11} & A_{12} \\
0 & U_S*P_RS
\end{bmatrix}
\]
... factoring the 3rd matrix.

... Note that \(P_LS*L_S\) is \(m-1\) by \(r'\) and \(U_S*P_RS\) is \(r'\) by \(n-1\)

\[
= (P'_L)^t * \begin{bmatrix}
1 & 0 \\
X & P_LS*L_S
\end{bmatrix} * \begin{bmatrix}
A_{11} & A_{12} \\
0 & U_S*P_RS
\end{bmatrix}
\]
... multiplying factors 2 and 3

\[
= (P'_L)^t * \begin{bmatrix}
1 & 0 \\
0 & P_LS
\end{bmatrix} * \begin{bmatrix}
1 & 0 \\
P_LS^{-t}*X & L_S
\end{bmatrix} * \begin{bmatrix}
A_{11} & A_{12} \\
0 & U_S*P_RS
\end{bmatrix}
\]
... factoring the 2nd matrix

... Note that \(L_S\) is \(m-1\) by \(r'\) and \(P_LS^{-t}*X\) is \(m-1\) by \(1\)

\[
= P_L * \begin{bmatrix}
1 & 0 \\
P_LS^{-t}*X & L_S
\end{bmatrix} * \begin{bmatrix}
A_{11} & A_{12} \\
0 & U_S*P_RS
\end{bmatrix}
\]
... multiplying factors 1 and 2, both permutation matrices,
... to get the final permutation matrix \(P_L\), by Lemma 2 part (4)
\[
= P_L \begin{bmatrix} 1 & 0 \\ P_LS^t \times X \ 0 & U_S \end{bmatrix} \begin{bmatrix} 0 & P_RS \end{bmatrix}
\]

... factoring the 3rd matrix

... Note that \( U_S \) is \( r' \) by \( n-1 \) and \( A_{12} \cdot P_RS^t \) is 1 by \( n-1 \)

\[
= P_L \begin{bmatrix} 1 & 0 \\ P_LS^t \times X \ 0 & U_S \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \cdot P_RS^t \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot (P'_R)^t
\]

... multiplying factors 4 and 5, both permutation matrices,
... to get the final permutation matrix \( P_R \), by Lemma 2 part (4)

\[
= P_L \begin{bmatrix} 1 & 0 \\ P_LS^t \times X \ 0 & U_S \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \cdot P_RS^t \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot P_R
\]

... the final factorization, by noticing that the 2nd matrix
... is lower triangular with 1s on the diagonal as desired,
... and the 3rd matrix is upper triangular with nonzeros
... on the diagonal, as desired.
... Note that \( L \) has \( r = r'+1 \) columns and \( U \) has \( r \) rows.

**Corollary 2:** Let \( A = P_L \cdot L \cdot U \cdot P_R \) be the above factorization
of the \( m \) by \( n \) nonzero matrix \( A \), where \( L \) (\( U \)) has \( r \) columns (rows). Then

1. \( \text{range}(A) = \text{span}(P_L \cdot L) \)
2. \( \text{rank}(A) = r \)
3. \( \text{nullity}(A) = n-r \)
4. \( \text{nullspace}(A) \) may be computed as follows.
   Write \( U = [U_1 \ , \ U_2] \) where \( U_1 \) is \( r \) by \( r \) (and so upper triangular)
and \( U_2 \) is \( r \) by \( n-r \). Then
   \[
   \text{nullspace}(A) = \text{span} \left( P_R \cdot t \begin{bmatrix} -U_1^{-1} \cdot U_2 \end{bmatrix} \begin{bmatrix} I^{n-r} \end{bmatrix} \right)
   \]
5. If \( A \) is invertible if and only if \( m = n = r \), in which case
   \[
   A^{-1} = P_R \cdot t \begin{bmatrix} -U^{-1} \cdot L^{-1} \cdot P_L \end{bmatrix}
   \]
6. The complete solution set of \( Ax = b \) may be computed from the entries
of the LU factorization of \( A \) (see below).

Proof: Write \( L = [L_1 \ 1] \) and \( U = [U_1 \ 1, \ U_2 \ 1] \)

so that \( L_1 \) is square and lower triangular
and \( U_1 \) is square and upper triangular. Since the diagonal entries
of \( L_1 \) and \( U_1 \) are nonzero, \( L_1 \) and \( U_1 \) are invertible, and have rank = \( r \).
Thus \( L \) has \( \text{nullity}(L) = 0 \) and so \( \text{rank}(L) = r \), since
\[
L \cdot x = 0 \Rightarrow [L_1 \cdot x] = 0 \Rightarrow [L_1 \cdot x] = 0 \Rightarrow x = 0 \Rightarrow L \text{ is one-to-one}
\]

(1): \( \text{range}(A) = \{ A \cdot x, \text{ all } x \text{ in } F^n \} \)
\[
\{P_L*L*U*P_R*x, \text{ all } x \text{ in } F^n\} = \{P_L*L*U*x, \text{ all } x \text{ in } F^n\}
\]

... since \(\{x: x \text{ in } F^n\} = \{P_R*x: x \text{ in } F^n\}\)

\[
\{P_L*L*[ x1 ], \text{ all } x1 \text{ in } F^r \text{ and } x2 \text{ in } F^{\{n-r}\}\}
\]

... by multiplying out

\[
\{P_L*L*(x1 + U2*x2), \text{ all } x1 \text{ in } F^r \text{ and } x2 \text{ in } F^{\{n-r}\}\}
\]

... since \(U1\) is invertible, \(\{x1: x1 \text{ in } F^r\} = \{U1*x1: x1 \text{ in } F^r\}\)

\[
\{P_L*L*x1, \text{ all } x1 \text{ in } F^r\}
\]

... since \(F^r = \{x1: x1 \text{ in } F^r\}\)

\[
\{x1 + U2*x2: x1 \text{ in } F^r \text{ and } x2 \text{ in } F^{\{n-r}\}\}
\]

= range(\(P_L*L\))

(2) \(\text{rank}(A) = \dim(\text{range}(A))\)

\[
= \dim(\text{range}(P_L*L))
= \dim(P_L*\text{range}(L))
= \dim(\text{range}(L)) \quad \text{... since } P_L \text{ is invertible}
= \text{rank}(L)
= r
\]

(3) By the dimension theorem, \(\text{nullity}(A) = \#\text{columns}(A) - \text{rank}(A) = n-r\)

(4) \(Ax=0\) implies that \(P_L*L*U*P_R*x = 0\), and that \(U*P_R*x = 0\), since \(\text{nullity}(L) = 0\). Let \(P_R*x = [x1]_r\)

\[
[x2]_{n-r}
\]

so that \(Ax = 0 => 0 = [U1 U2]*[x1] = U1*x1+U2*x2\).

\[
[x2]
\]

Since \(U1\) is invertible, this means \(x1 = -U1^{-1}*U2*x2\).

In other words, a vector \(x\) is in \(\text{nullspace}(A)\) iff \(x = P_R^\top * [x1] = P_R^\top * [-U1^{-1}*U2] * x2, \text{ for any } x2 \text{ in } F^{\{n-r\}}\)

\[
[x2]_{n-r}
\]

or \(x \text{ in } \text{span}( P_R^\top * [-U1^{-1}*U2] )\)

\[
[x2]_{n-r}\]

(5) \(A\) is invertible iff it is square (m=n) and has \(\text{rank}(A) = r = n\).

In this case \(A^{-1} = (P_L * L * U * P_R)^{-1}\)

\[
= P_R^\top * [-U1^{-1}*U2] * L^\top * [-U1^{-1}]*P_L^\top
= P_R^\top * U^\top * L^\top * P_L^\top
\]

(6) We can solve \(Ax=b\) if \(b\) in \(\text{range}(A) = \text{range}(P_L*L)\), i.e.

if \(P_L^\top t*b\) in \(\text{range}(L) = \text{range}([L1])\) or \(P_L^\top t*b = [L1]*x' = [L1*x']\)
for some $x'$. Write $P_L \cdot t \cdot b = [b_1] \cdot r$ so $b_1 = L_1 \cdot x'$ and $b_2 = L_2 \cdot x'$

Thus $x' = L_1^{-1} \cdot b_1$ and $b_2 = L_2 \cdot L_1^{-1} \cdot b_1$.
In other words, we can solve $Ax = b$ if and only if $b_2 = L_2 \cdot L_1^{-1} \cdot b_1$

where $P_L \cdot t \cdot b = [b_1] \cdot r$.

If this condition holds, we can find all the solutions as follows. Suppose there are two solutions, $x$ and $y$. Then

$$A \cdot (x - y) = A \cdot x - A \cdot y = b - b = 0$$

so $x - y$ must be in nullspace($A$). In other words, if we find one solution $x$, all the others can be found by taking $x$ plus any member of nullspace($A$) from part (4). To find one solution $x$, we solve

$$x' = U \cdot P_R \cdot x$$
for $x$ or

$$x' = [U_1, U_2] \cdot P_R \cdot x = [U_1, U_2] \cdot [x_1, x_2] = U_1 \cdot x_1 + U_2 \cdot x_2$$

One solution is gotten by setting $x_2 = 0$, and solving $x' = U_1 \cdot x_1$
for $x_1 = U_1^{-1} \cdot L_1^{-1} \cdot b_1$. Thus the
complete solution set of $Ax = b$, if a solution exists at all, is

$$\{ x = P_R \cdot t \cdot [U_1, U_2] \cdot L_1^{-1} \cdot b_1 + z, \text{ for all } z \text{ in nullspace}(A) \}$$

Finally, we consider the number of arithmetic operations required to compute the LU factorization. We note that multiplying by permutation matrices does not require any arithmetic operations, only reordering rows (or columns). We assume for simplicity that $A$ is $n \times n$ with rank $n$ (if the rank is lower the number of operations is lower too). Let $C(n)$ be the "cost", i.e. number of operations. Then by the proof of Theorem 2

$$C(n) = \text{number of operations to compute } X = A_{21}/a_{11}$$
$$+ \text{number of operations to compute } S = A_{22} - X \cdot A_{12}$$
$$+ \text{number of operations to compute the LU factorization of } S$$
$$= n-1 \text{ divisions}$$
$$+ (n-1)^2 \text{ multiplications to form } X \cdot A_{12}$$
$$+ (n-1)^2 \text{ substractions to subtract } A_{22} - X \cdot A_{12}$$
$$+ C(n-1)$$

This gives us the recurrence

$$C(n) = 2 \cdot (n-1)^2 + (n-1) + C(n-1)$$
along with the initial value $C(1) = 0$, or

$$C(n) = \text{sum}_{j=1 \text{ to } n-1} (2 \cdot j^2 + j)$$
$$= 2/3 \cdot n^3 + "\text{lower order terms}"$$
where "lower order terms" means a quadratic function of $n$. When $n$ is large, $C(n)$ is well approximated by $\frac{2}{3}n^3$.

It is interesting to note that solving a system of linear equations costs $\frac{2}{3}n^3 + \text{lower order terms}$, a third as much as multiplying two matrices of the same size, $2n^3$. 