3.1.1

Because $u(n,n)=0$ and $z \neq 0$, we can specify $z(n)$ to any non zero scalar, I'll use 1 in my code.

```
function [z] = unull(U)
n= size(U,1);
z= zeros(n,1);
z(n)=1;
for i=n-1:-1:1
    z(i)= (z(i)- U(i,i+1:n) * z(i+1:n)) / U(i,i);
end
end
```

3.1.7 $LXK = B$

$LXK = B \Leftrightarrow L(XK) = B$

We can treat XK as a whole $n \times n$ matrix and solve it using Multiple right hand sides algorithm, which is a block forward elimination scheme described in book. Then we get $XK = C \in \mathbb{R}^{n \times n}$.

Then, transpose both sides, $K^T X^T = C^T$, K^T is an upper triangular, and we can solve it using a block backward substitution which is entirely analogous to the previous one. Therefore, we solve X^T, transpose X^T and get X in the end.

P3.2.3

① Show that after r steps of outer product LU, $A(r+1:n, r+1:n)$ houses S.

We have, $S = A_{22} - L_{21} U_{12}$

In outer product LU algorithm,

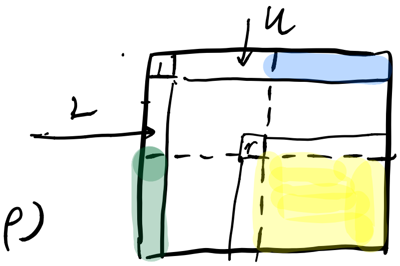
for $k = 1 : n-1$

$\quad \rho = k+1 : n$

$\quad A(\rho, k) = A(\rho, k) / A(k, k)$

$\quad A(\rho, \rho) = A(\rho, \rho) - A(\rho, k) \cdot A(k, \rho)$

end



For $A(r+1:n, r+1:n)$, this part will be subtracted by $A(r+1:n, k) \cdot A(k, r+1:n)$. in the kth iteration, where $A(r+1:n, k)$ is overwritten by $L_{21}(:, k)$ and $A(k, r+1:n)$ is overwritten by $U_{12}(k, :)$ already.

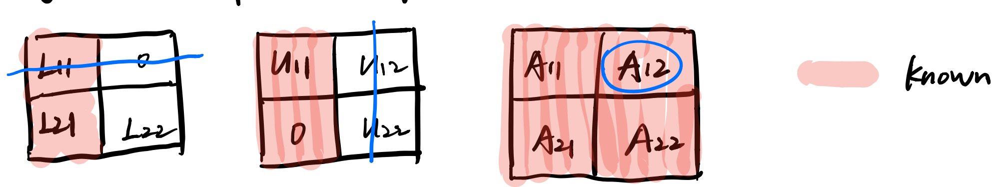
Thus, after r steps,

$$A(r+1:n, r+1:n) = A_{22} - L_{21}(:, 1:r) \cdot U_{12}(1:r, :)$$

$$= A_{22} - L_{21} \cdot U_{12} = S. \qquad \text{It houses } S.$$

② How could S be obtained after r steps of Gaxpy LU?

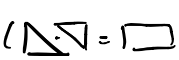
Because Gaxpy LU is a left-looking algorithm, after r steps, we can only get the left part of L, U, like this:



In $S = A_{22} - L_{21} U_{12}$, we still need U_{12} to obtain S.

So based on our block matrices,

We have $\quad L_{11} \cdot U_{12} + 0 \cdot U_{22} = A_{12}$

$\qquad L_{11} \cdot U_{12} = A_{12}$ () A_{12}, L_{11} are known.

We can solve $L_{11}U_{12} = A_{12}$ by multiple RHS algorithm to get U_{12}.

Then, $S = A_{22} - L_{21} \cdot U_{12}$ can be obtained.

P3.2.9

$$\begin{array}{c} \\ n-1 \\ 1 \\ \\ \end{array} \begin{array}{cc} & \\ \left(\begin{array}{cc} L_{11} & 0 \\ l_{21} & 1 \end{array}\right) \\ n-1 \quad 1 \end{array} \cdot \begin{array}{c} n-1 \quad 1 \\ \left(\begin{array}{cc} U_{11} & u_{12} \\ 0 & u_{22} \end{array}\right) \\ \\ \end{array} = \begin{array}{c} \\ \left(\begin{array}{cc} A_{11} & a_{12} \\ a_{21} & a_{22} \end{array}\right) \\ n-1 \quad 1 \end{array} \begin{array}{c} \\ n-1 \\ 1 \\ \\ \end{array}$$

① Recursive version:

```
function [L,U] = dotlu(A)
n= size(A,1);
U= zeros(n); L= eye(n);
A11 = A(1:n-1, 1:n-1);
[L11, U11]= dotlu(A11);
L(1:n-1, 1:n-1) = L11;
U(1:n-1, 1:n-1) = U11;
ã= A(:, n);
Solve L(1:n-1,1:n-1)·z = ã(1:n-1) for z∈R^{n-1}
U(1:n-1, n)= z;
b̃= A(n, :);
Solve z·U(1:n-1,1:n-1) = b̃(1:n-1) for z^T∈R^{n-1}.
L(n, 1:n-1) = z;
U(n,n) = A(n,n)- L(n,1:n-1)* U(1:n-1, n);        % dot product.
end
```

② Iterative version:

Initialize L to the Identity and U to the zero matrix.

```
for j=1:n-1
  if j=1
      U(1,1)= A(1,1);
  else
    ã= A(:,j);
    Solve L(1:j-1,1:j-1)·z= ã(1:j-1) for z∈R^{j-1}.
    U(1:j-1, j) = z;
    b̃= A(j,:);
    Solve z·U(1:j-1,1:j-1)= b̃(1:j-1) for z^T∈R^{j-1}.
    L(j,1:j-1)= z;
    for k=1:j-1
       U(j,j)=A(j,j)-L(j,k)* U(k,j); % dot product
    end
  end
```