

Continuing with pivoted LU factorization

$$LU = PA$$

Thus

$$A = P^{-1}LU$$

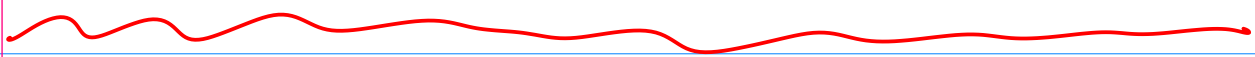
matrix mult. is composition of linear functions.

parenthesis mean compute LU first and then apply P^{-1} .

```
julia> (L*U)[invperm(p),:]  
3x3 Matrix{Float64}:  
1.0 2.0 3.0  
4.0 5.0 6.0  
1.0 0.0 1.0
```

$$P^{-1}(L(U(z)))$$

compute the LU first... i.e. means apply the functions from right to left...



```
PA = julia> A[p,:]  
3x3 Matrix{Int64}:  
4 5 6  
1 0 1  
1 2 3  
= LU = julia> L*U  
3x3 Matrix{Float64}:  
4.0 5.0 6.0  
1.0 0.0 1.0  
1.0 2.0 3.0
```

When we did the factorization using the commands →

```
julia> A  
3x3 Matrix{Int64}:  
1 2 3  
4 5 6  
1 0 1  
  
julia> L,U,p=lu(A)  
LU{Float64, Matrix{Float64}}  
L factor:  
3x3 Matrix{Float64}:  
1.0 0.0 0.0  
0.25 1.0 0.0  
0.25 -0.6 1.0  
U factor:  
3x3 Matrix{Float64}:  
4.0 5.0 6.0  
0.0 -1.25 -0.5  
0.0 0.0 1.2
```

Cholesky factorization:

If $A \in \mathbb{R}^{n \times n}$ and $A^T = A$

Symmetric

and $x \cdot Ax > 0$ for every $x \in \mathbb{R}^n$

positive definite means no pivoting need to make LU factorization..

Then there is a triangular matrix L such that

$$A = LL^T$$

L is a lower triangular matrix.

This is like LU factorization where $U = L^T$.

Suppose $A = LU$ Can we find the Cholesky factorization from the LU factorization? Seems similar..

$$A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix}$$

```
julia> B=[4 2 1; 2 3 1; 1 1 4]
3x3 Matrix{Int64}:
 4  2  1
 2  3  1
 1  1  4
```

This matrix is symmetric and positive definite.

```
julia> L,U,p=lu(B)
LU{Float64, Matrix{Float64}}
L factor:
3x3 Matrix{Float64}:
 1.0  0.0  0.0
 0.5  1.0  0.0
 0.25 0.25 1.0
U factor:
3x3 Matrix{Float64}:
 4.0  2.0  1.0
 0.0  2.0  0.5
 0.0  0.0  3.625

julia> p
3-element Vector{Int64}:
 1
 2
 3
```

no permutation was used in this case...

Try the LU factorization

$$U = \begin{bmatrix} 4 & 2 & 1 \\ 0 & 2 & .5 \\ 0 & 0 & 3.625 \end{bmatrix}$$

diagonal matrix

$$\begin{aligned} r_1 &\leftarrow \frac{1}{4}r_1 \\ r_2 &\leftarrow \frac{1}{2}r_2 \\ r_3 &\leftarrow \frac{1}{3.625}r_3 \end{aligned}$$

```
julia> U[1,:]=1/4*U[1,:];
U[2,:]=1/2*U[2,:];
U[3,:]=1/3.625*U[3,:];

julia> U
3x3 Matrix{Float64}:
 1.0  0.5  0.25
 0.0  1.0  0.25
 0.0  0.0  1.0
```

This really is L^T

$$D = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3.625 \end{bmatrix}$$

undoes the rescaling

Thus...

$$A = LU = LDL^T$$

$$= \begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix} \text{ same} \leftrightarrow$$

```
julia> D=diagm([4,2,3.625])
3x3 Matrix{Float64}:
 4.0  0.0  0.0
 0.0  2.0  0.0
 0.0  0.0  3.625
```

```
julia> L*D*L'
3x3 Matrix{Float64}:
 4.0  2.0  1.0
 2.0  3.0  1.0
 1.0  1.0  4.0
```

$$\sqrt{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & \sqrt{3.625} \end{bmatrix}$$

note $\sqrt{D} = \sqrt{D}^T$

$$A = LU = LDL^T = L\sqrt{D}\sqrt{D}L^T$$

$$= L\sqrt{D}\sqrt{D}^T L^T = (L\sqrt{D})(L\sqrt{D})^T$$

Cholesky factor

verification

```
julia> L2=L*sqrt(D)
3x3 Matrix{Float64}:
 2.0  0.0  0.0
 1.0  1.41421  0.0
 0.5  0.353553  1.90394
```

```
julia> L2*L2'
3x3 Matrix{Float64}:
 4.0  2.0  1.0
 2.0  3.0  1.0
 1.0  1.0  4.0
```

So LU factorization can be used to find the Cholesky decomposition. However, some computational efficiency results if we numerically aim for the Cholesky from the beginning...next time