Upper Triangularizing

Problem : Find Gauss transformations $\mathbf{M}_1, \ldots, \mathbf{M}_{n-1} \in \mathbb{R}^{n\times n}$ such that

$$\mathbf{M}_{n-1}\cdots\mathbf{M}_2\mathbf{M}_1\mathbf{A} = \mathbf{U}, \quad \mathbf{U} \text{ is upper triangular}$$

Step 1: Choose $\mathbf{M}_1$ s.t. $\mathbf{M}_1\mathbf{a}_1 = [\ a_{11}, 0, \ldots, 0\ ]^T$

- If $a_{11} \neq 0$, let

$$\mathbf{M}_1 = \mathbf{I} - \boldsymbol{\tau}^{(1)}\mathbf{e}_1^T, \qquad \boldsymbol{\tau}^{(1)} = [\ 0, a_{21}/a_{11}, \ldots, a_{n1}/a_{11}\ ]^T.$$

$$\mathbf{M}_1\mathbf{A} = \begin{array}{c} \begin{array}{cccc} \phantom{a_{11}} & a_{12} & & a_{1n} \end{array} \\ \begin{bmatrix} a_{11} & \times & \ldots & \times \\ 0 & \times & \ldots & \times \\ \vdots & \vdots & & \vdots \\ 0 & \times & \ldots & \times \end{bmatrix} \end{array}$$

Upper Triangularizing (cont'd)

Step 2: Set $\mathbf{A}^{(1)} = \mathbf{M}_1\mathbf{A}$

Choose $\mathbf{M}_2$ s.t. $\mathbf{M}_2\mathbf{a}_2^{(1)} = [\ a_{12}^{(1)}, a_{22}^{(1)}, 0, \ldots, 0\ ]^T$

2nd column of $A^{(1)}$

- If $a_{22}^{(1)} \neq 0$, let

$$\mathbf{M}_2 = \mathbf{I} - \boldsymbol{\tau}^{(2)}\mathbf{e}_2^T, \qquad \boldsymbol{\tau}^{(2)} = [\ 0, 0, a_{32}^{(1)}/a_{22}^{(1)}, \ldots, a_{n,2}^{(1)}/a_{22}^{(1)}\ ]^T$$

$$\mathbf{M}_2\mathbf{A}^{(1)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \times & \ldots & \times \\ 0 & a_{22}^{(1)} & \times & \ldots & \times \\ \vdots & 0 & \times & & \times \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \times & \ldots & \times \end{bmatrix}$$

- Applying $\mathbf{M}_2$ to $\mathbf{A}^{(1)}$ does not change the first column of $\mathbf{A}^{(1)}$

$$\mathbf{M}_2\mathbf{a}_1^{(1)} = \mathbf{a}_1^{(1)} - \boldsymbol{\tau}^{(2)} \underbrace{a_{21}^{(1)}}_{=0} = \mathbf{a}_1^{(1)}$$

$$= (I - \tau^{(2)} e_2^T) a_1^{(1)} = a_1^{(1)} - \tau^{(2)} (e_2^T a_1^{(1)})$$

Upper Triangularizing (cont'd)

Let $\mathbf{A}^{(0)} = \mathbf{A}$ and $\mathbf{A}^{(k)} = \mathbf{M}_k \mathbf{A}^{(k-1)} = \mathbf{M}_k \cdots \mathbf{M}_2 \mathbf{M}_1 \mathbf{A}$

Step k: Choose $\mathbf{M}_k$ s.t. $\mathbf{M}_k \mathbf{a}_k^{(k-1)} = \left[a_{1k}^{(k-1)}, \ldots, a_{kk}^{(k-1)}, 0, \ldots, 0\right]^T$

- If $a_{kk}^{(k-1)} \neq 0$, let

$$\mathbf{M}_k = \mathbf{I} - \boldsymbol{\tau}^{(k)} \mathbf{e}_k^T, \qquad \boldsymbol{\tau}^{(k)} = \left[0, \ldots, 0, \frac{a_{k+1,k}^{(k-1)}}{a_{kk}^{(k-1)}}, \ldots, \frac{a_{n,k}^{(k-1)}}{a_{kk}^{(k-1)}}\right]^T,$$

$$\mathbf{A}^{(k)} = \mathbf{M}_k \mathbf{A}^{(k-1)} = \begin{bmatrix} a_{11}^{(k-1)} & \cdots & a_{1k}^{(k-1)} & \times & \cdots & \times \\ 0 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & & a_{kk}^{(k-1)} & \vdots & & \vdots \\ \vdots & & 0 & \times & & \times \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \times & \cdots & \times \end{bmatrix}$$

changed

- Applying $\mathbf{M}_k$ to $\mathbf{A}^{(k-1)}$ does not change the first $k-1$ columns of $\mathbf{A}^{(k-1)}$
- $\mathbf{A}^{(n-1)} = \mathbf{U}$ is upper triangular

Upper Triangularizing (cont'd)

Example: Upper triangularize $\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix}$

$$M_1 = I - \tau^{(1)} e_1^T = \begin{bmatrix} 1 & & \\ -2 & 1 & \\ -3 & & 1 \end{bmatrix} \qquad \tau^{(1)} = \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix}$$

$$M_1 A = \begin{bmatrix} 1 & 4 & 7 \\ 0 & \boxed{-3 \quad -6} \\ 0 & \boxed{-6 \quad -11} \end{bmatrix} \begin{matrix} \rightsquigarrow [5 \quad 8] - 2[4 \quad 7] \\ \rightsquigarrow [6 \quad 10] - 3[4 \quad 7] \end{matrix}$$

$$\| \quad A^{(1)}$$

$$M_2 = I - \tau^{(2)} e_2^T = \begin{bmatrix} 1 & & \\ & 1 & \\ & -2 & 1 \end{bmatrix} \qquad \tau^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

$$A^{(2)} = M_2 A^{(1)} = \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & \boxed{1} \end{bmatrix} \rightsquigarrow -11 - 2 \times (-6)$$

Computation of $\mathbf{L}$

When the pivots $a_{kk}^{(k-1)} \neq 0$ for all $k = 1, \ldots, n-1$,

$$\mathbf{U} = \mathbf{M}_{n-1} \cdots \mathbf{M}_2 \mathbf{M}_1 \mathbf{A} \text{ is upper triangular}$$

Find $\mathbf{L}$ based on $\mathbf{U}$

Facts: Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ be two lower (upper) triangular matrices. Then,

1. $\mathbf{AB}$ is lower (upper) triangular. In addition, if $\mathbf{A}$, $\mathbf{B}$ have unit diagonal entries, then $\mathbf{AB}$ is unit lower (upper) triangular.
2. $\det(\mathbf{A}) = \prod_{i=1}^{n} a_{ii}$.
3. If $\mathbf{A}$ is nonsingular, $\mathbf{A}^{-1}$ is lower (upper) triangular with $[\mathbf{A}^{-1}]_{ii} = 1/a_{ii}$.

Fact 2 $\Downarrow$

Fact 3 $\Rightarrow$ M_k^{-1} unit lower triangular

Since every $\mathbf{M}_k$ is unit lower triangular (invertible),

$$\mathbf{L} = \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \cdots \mathbf{M}_{n-1}^{-1}$$

Fact 1 $\Rightarrow$ L unit lower triangular

satisfies $\mathbf{A} = \mathbf{LU}$ and is unit lower triangular

Proof of Facts

Fact 1. Let A, B be lower triangular.

For simplicity, let $C = A^T$ and $D = AB = C^T B$.

$$\Updownarrow$$

$$\boxed{d_{k\ell} = c_k^T b_\ell}$$

Since B is lower triangular,

$$b_\ell = \sum_{j=\ell}^{n} b_{j\ell} e_j, \quad \ell = 1, \dots, n$$

Since $C = A^T$ is upper triangular, $c_k = \sum_{i=1}^{k} c_{ik} e_i$. $\quad k = 1, \dots, n$

Thus, $d_{k\ell} = \Big(\sum_{i=1}^{k} \underset{\underset{a_{ki}}{\parallel}}{c_{ik}} e_i \Big)^T \Big(\sum_{j=\ell}^{n} b_{j\ell} e_j \Big)$

$$= \sum_{i=1}^{k} \sum_{j=\ell}^{n} a_{ki} b_{j\ell} e_i^T e_j .$$

Note that

$$e_i^T e_j = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

Proof of Facts (cont'd)

Therefore, $d_{kl} = \begin{cases} 0 & l > k \\ \sum_{j=l}^{k} a_{kj} b_{jl} & l \le k \end{cases}$

$\Rightarrow D$ is lower triangular.

If A and B are unit lower triangular. $a_{ii} = b_{ii} = 1$ $\forall i = 1, \cdots, n$.

Hence, $d_{ii} = a_{ii} b_{ii} = 1 \Rightarrow D$ unit lower triangular.

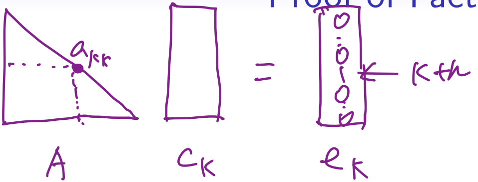
Fact 3 Let A be nonsingular and lower triangular.

According to Fact 2, $a_{kk} \neq 0$ $\forall k = 1, \cdots, n$.

Let $C = A^{-1}$. Then,

$AC = I \iff Ac_k = e_k, \quad k = 1, \cdots, n$

Proof of Facts (cont'd)



$$
\begin{aligned}
&a_{11} c_{1k} &&= 0 \\
&a_{21} c_{1k} + a_{22} c_{2k} &&= 0 \\
&\vdots && \\
&a_{k-1,1} c_{1k} + \cdots + a_{k-1,k-1} c_{k-1,k} &&= 0 \\
&a_{k,1} c_{1k} + \cdots + a_{kk} c_{kk} &&= 1
\end{aligned}
$$

$\Rightarrow c_{1k} = \cdots = c_{k-1,k} = 0$ and $c_{kk} = \dfrac{1}{a_{kk}}$

Therefore, C is lower triangular with $c_{kk} = \dfrac{1}{a_{kk}} \ \forall k$

A Naive Implementation of LU (Don't Use It)

n×n identity matrix

n-dim zero vector

$A^{(k-1)}$

```
function [L,U]= naive_LU(A)
n= size(A,1);
L= eye(n); tau= zeros(n,1); U= A;
for k=1:n-1,
     rows= k+1:n;
     tau(rows)= U(rows,k)/U(k,k);
     M= eye(n); M(rows,k)= -tau(rows);
     U= M*U;            % compute A^(k) = M_k A^(k-1)
     L= L*inv(M);
    % to eventually obtain L = M_1^-1 M_2^-1 ... M_{n-1}^-1
end
```

- The code treats each $\mathbf{A}^{(k)} = \mathbf{M}_k \mathbf{A}^{(k-1)}$ as a general matrix multiplication process, requiring $\mathcal{O}(n^3)$ flops. Can we utilize the structure of $\mathbf{M}_k$ to reduce complexity?
- The code calls for $n-1$ matrix inversion to compute $\mathbf{L}$. Why not directly compute the inverse of $\mathbf{A}$?

$$L = I + \begin{bmatrix} \tau^{(1)} \end{bmatrix} + \begin{bmatrix} \tau^{(2)} \end{bmatrix} + \cdots + \begin{bmatrix} \tau^{(n)} \end{bmatrix}$$

Computation of L (cont'd)

$$M_k = I - \tau^{(k)} e_k^T$$

no overlap among nonzero elements $\Rightarrow$ 0 flops

> **Fact**: $\mathbf{M}_k^{-1} = \mathbf{I} + \boldsymbol{\tau}^{(k)}\mathbf{e}_k^T$ for each $k = 1, \ldots, n-1$

Verification: Since $[\boldsymbol{\tau}^{(k)}]_k = 0$,

$$\begin{aligned}(\mathbf{I} + \boldsymbol{\tau}^{(k)}\mathbf{e}_k^T)\mathbf{M}_k &= (\mathbf{I} + \boldsymbol{\tau}^{(k)}\mathbf{e}_k^T)(\mathbf{I} - \boldsymbol{\tau}^{(k)}\mathbf{e}_k^T) \\ &= \mathbf{I} + \boldsymbol{\tau}^{(k)}\mathbf{e}_k^T - \boldsymbol{\tau}^{(k)}\mathbf{e}_k^T + \boldsymbol{\tau}^{(k)}\underbrace{\mathbf{e}_k^T\boldsymbol{\tau}^{(k)}}_{=0}\mathbf{e}_k^T = \mathbf{I}\end{aligned}$$

$$M_1^{-1} M_2^{-1} M_3^{-1} = (I + \tau^{(1)} e_1^T)(I + \tau^{(2)} e_2^T)(I + \tau^{(3)} e_3^T)$$

Using the same spirit,

$$= (I + \tau^{(1)} e_1^T + \tau^{(2)} e_2^T + \tau^{(1)} \underbrace{e_1^T \tau^{(2)}}_{=0} e_2^T) \cdot (I + \tau^{(3)} e_3^T)$$

$$\mathbf{L} = \mathbf{M}_1^{-1} \ldots \mathbf{M}_{n-1}^{-1} = \mathbf{I} + \sum_{k=1}^{n-1} \boldsymbol{\tau}^{(k)}\mathbf{e}_k^T$$

$$= I + \tau^{(1)} e_1^T + \tau^{(2)} e_2^T + \tau^{(3)} e_3^T + \tau^{(1)} \underbrace{e_1^T \tau^{(3)}}_{=0} e_3^T + \tau^{(2)} \underbrace{e_2^T \tau^{(3)}}_{=0} e_3^T$$

An Improved LU Code (Still Not Used by MATLAB)

```
function [L,U]= better_LU(A)
n= size(A,1);
L= eye(n); tau= zeros(n,1); U= A;
for k=1:n-1,
     rows= k+1:n;
     tau(rows)= U(rows,k)/U(k,k);
     U(rows,rows)= U(rows,rows)- tau(rows)*U(k,rows);
     U(rows,k)= 0;
     L(rows,k)= tau(rows);
end
```

$A^{(k)}$

$(n-k)^2$ multiplications

$(n-k)^2$ subtractions

no flops

- Complexity: $\mathcal{O}(2n^3/3)$
- Again, need nonzero pivots $a_{kk}^{(k-1)}$

$$\sum_{k=1}^{n-1} 2(n-k)^2 \left(+\,(n-k)\right) \quad \text{ignore}$$

$$= \cancel{2\sum_{k=1}^{n-1} n^2} - \cancel{4n\sum_{k=1}^{n-1} k} + 2\sum_{k=1}^{n-1} k^2 = O\left(\frac{2}{3}n^3\right)$$