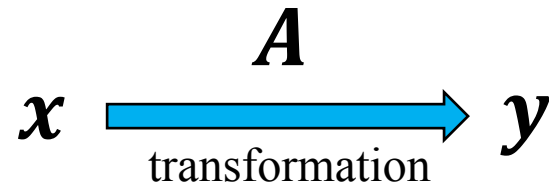


# Solving Linear System of Equations

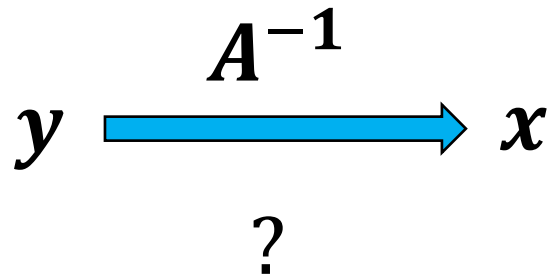
# The “Undo” button for Linear Operations

Matrix-vector multiplication: given the data  $\mathbf{x}$  and the operator  $\mathbf{A}$ , we can find  $\mathbf{y}$  such that

$$\mathbf{y} = \mathbf{A} \mathbf{x}$$



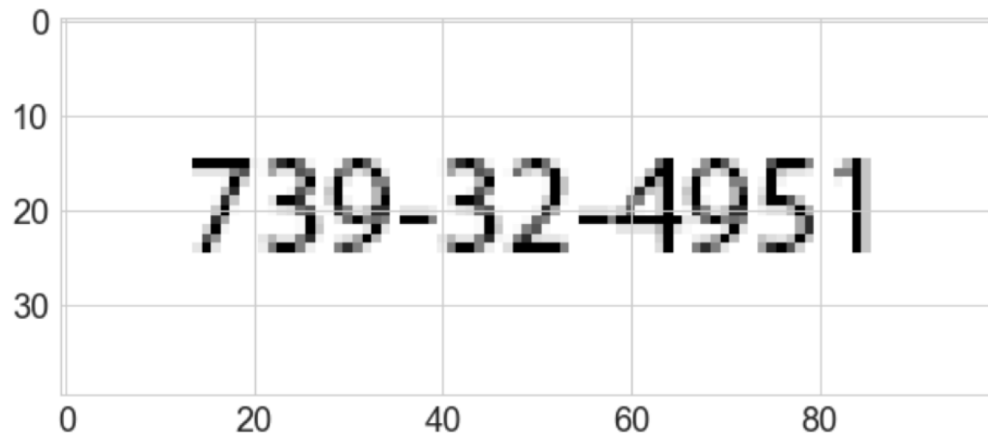
What if we know  $\mathbf{y}$  but not  $\mathbf{x}$ ? How can we “undo” the transformation?



Solve  $\mathbf{A} \mathbf{x} = \mathbf{y}$  for  $\mathbf{x}$

# Image Blurring Example

Demo “Image Blur Inverse”

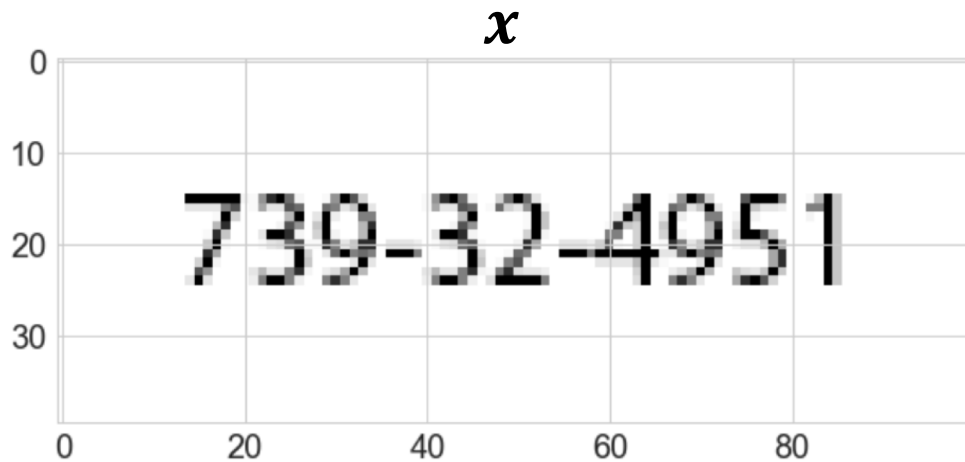


- Image is stored as a 2D array of real numbers between 0 and 1 (0 represents a white pixel, 1 represents a black pixel)
- ***xmat*** contains the 2D data (the image) with dimensions 100x40
- Flatten the 2D array as a 1D array
- ***x*** contains the 1D data with dimension 4000,
- Apply blurring operation to data ***x***, i.e.

$$\mathbf{y} = \mathbf{A} \mathbf{x}$$

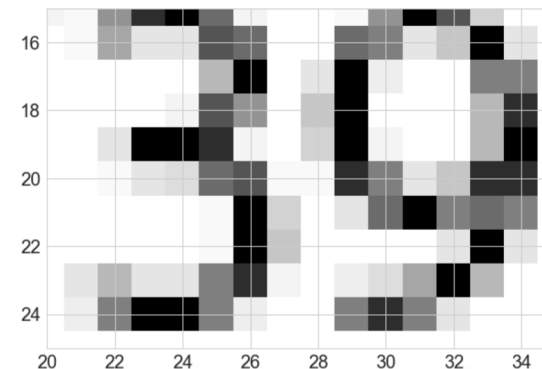
where ***A*** is the blur operator and ***y*** is the blurred image

# Blur operator



```
plt.imshow(xmat);  
plt.axis([20,35,25,15])
```

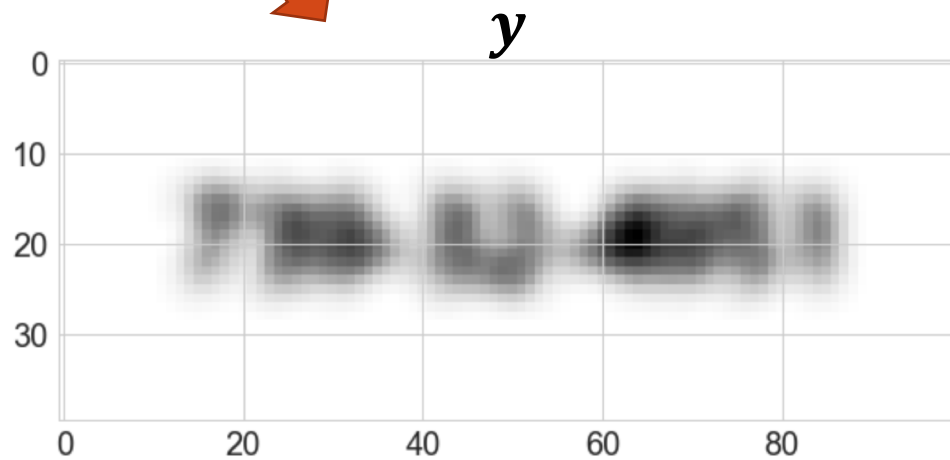
[20, 35, 25, 15]



blurred image (4000,)  
Blur operator (4000,4000)  
"original" image (4000,)

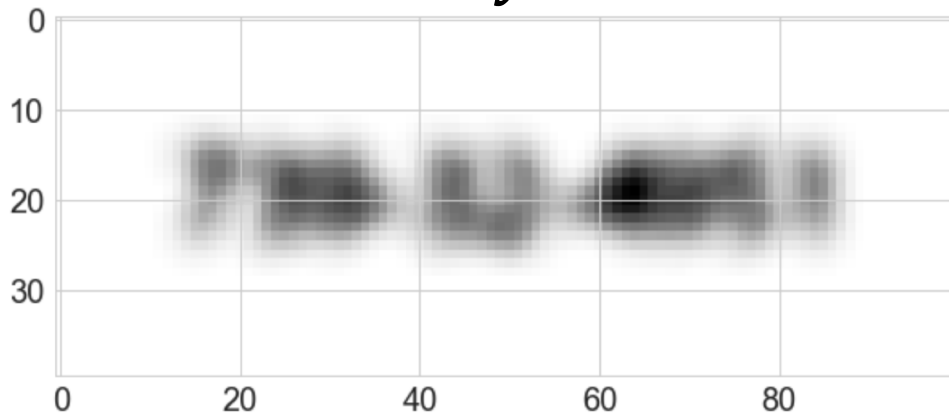
$$y = Ax$$

Blur operator  
 $A$



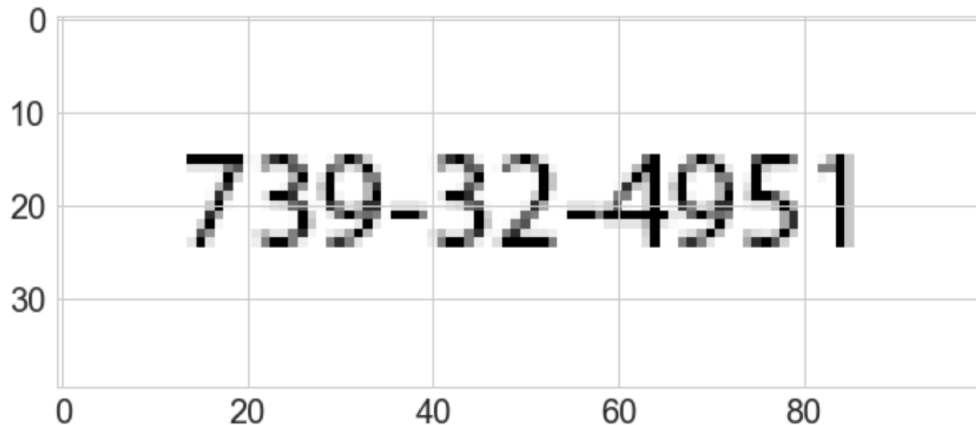
# ”Undo” Blur to recover original image

$y$



Solve  
 $Ax = y$   
for  $x$

$x$



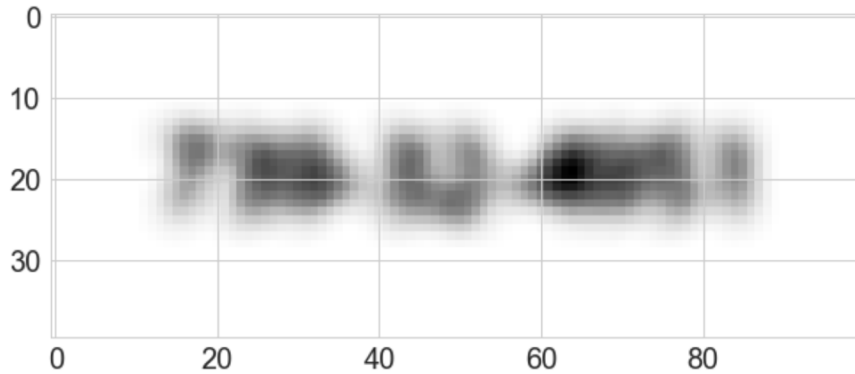
Assumptions:

1. we know the blur operator  $A$
2. the data set  $y$  does not have any noise (“clean data”)

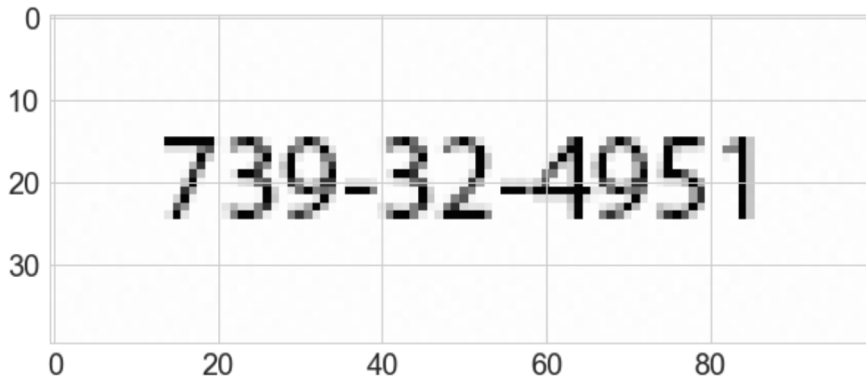
What happens if we add some noise to  $y$ ?

# ”Undo” Blur to recover original image

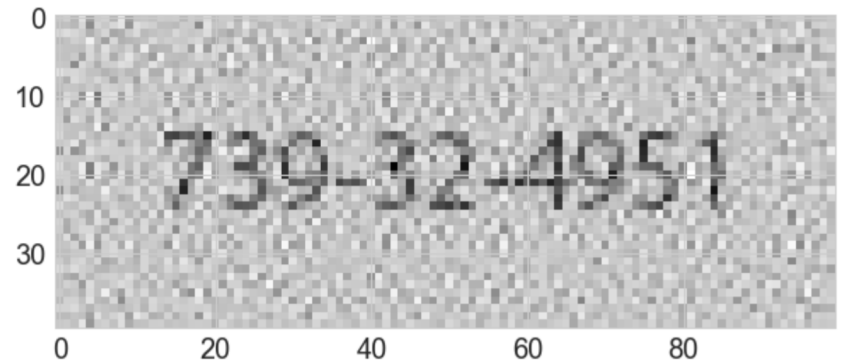
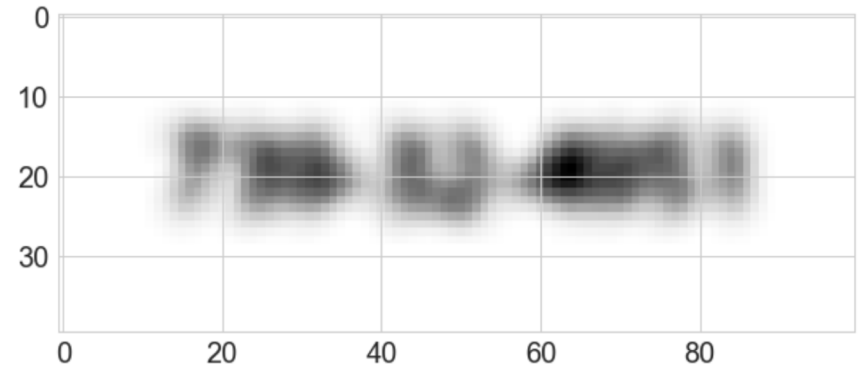
$$\mathbf{y} + a * 10^{-6} \quad (a \in (0,1))$$



Solve  $A \mathbf{x} = \mathbf{y}$  for  $\mathbf{x}$



$$\mathbf{y} + a * 10^{-4} \quad (a \in (0,1))$$



How much noise can we add and still be able to recover meaningful information from the original image? At which point this inverse transformation fails?

We will talk about sensitivity of the “undo” operation later.

# Linear System of Equations

How do we actually solve  $\mathbf{A} \mathbf{x} = \mathbf{b}$  ?

We can start with an “easier” system of equations...

Let's consider triangular matrices (lower and upper):

$$\begin{pmatrix} L_{11} & 0 & \dots & 0 \\ L_{21} & L_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \dots & L_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$\begin{pmatrix} U_{11} & U_{12} & \dots & U_{1n} \\ 0 & U_{22} & \dots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & U_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

# Example: Forward-substitution for lower triangular systems

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ 1 & 2 & 6 & 0 \\ 1 & 3 & 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 6 \\ 4 \end{pmatrix}$$



# Example:

$$\begin{pmatrix} 2 & 8 & 4 & 2 \\ 0 & 4 & 4 & 3 \\ 0 & 0 & 6 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 4 \\ 1 \end{pmatrix}$$

$$x_4 = \frac{1}{2}$$

$$x_3 = \frac{4 - 2\frac{1}{2}}{6} = \frac{1}{2}$$

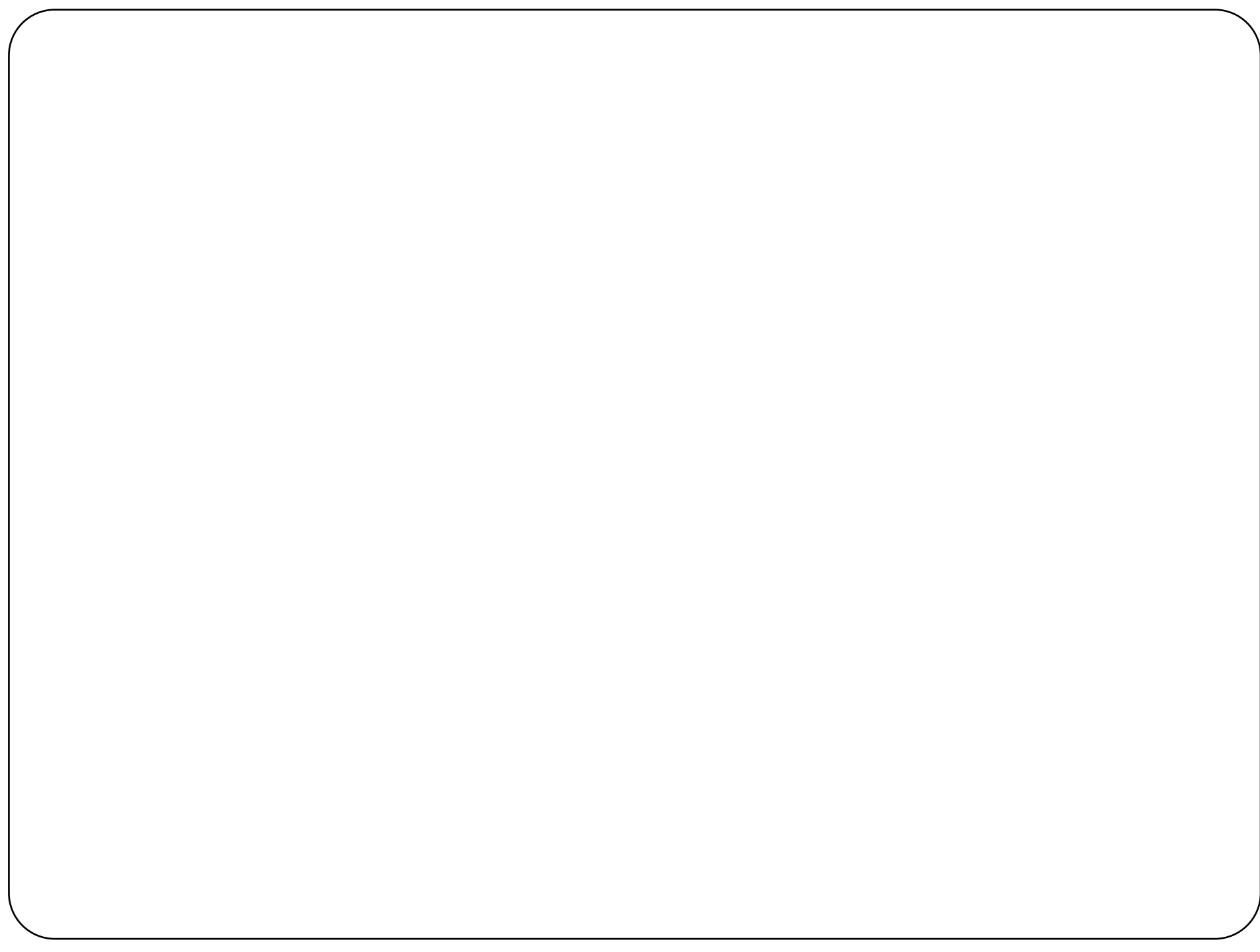
$$x_2 = \frac{4 - 4\frac{1}{2} - 3\frac{1}{2}}{4} = \frac{1/2}{4} = \frac{1}{8}$$

$$x_1 = \frac{2 - 8\frac{1}{8} - 4\frac{1}{2} - 2\frac{1}{2}}{2} = \frac{-2}{2} = -1$$

# Triangular Matrices

$$\begin{pmatrix} U_{11} & U_{12} & \cdots & U_{1n} \\ 0 & U_{22} & \cdots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & U_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Recall that we can also write  $\mathbf{U} \mathbf{x} = \mathbf{b}$  as a linear combination of the columns of  $\mathbf{U}$



# Triangular Matrices

Forward-substitution for lower-triangular systems:

$$\begin{pmatrix} L_{11} & 0 & \dots & 0 \\ L_{21} & L_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \dots & L_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$x_1 = b_1/L_{11} \qquad x_i = \frac{b_i - \sum_{j=1}^{i-1} L_{ij}x_j}{L_{ii}}, \qquad i = 2, 3, \dots, n$$

# Cost of solving triangular systems

$$x_n = b_n/U_{nn} \quad x_i = \frac{b_i - \sum_{j=i+1}^n U_{ij}x_j}{U_{ii}}, \quad i = n-1, n-2, \dots, 1$$

# Cost of solving triangular systems

$$x_n = b_n / U_{nn} \quad x_i = \frac{b_i - \sum_{j=i+1}^n U_{ij} x_j}{U_{ii}}, \quad i = n-1, n-2, \dots, 1$$

$n$  divisions

$n(n-1)/2$  subtractions/additions

$n(n-1)/2$  multiplications



Computational complexity is  $O(n^2)$

$$x_1 = b_1 / L_{11} \quad x_i = \frac{b_i - \sum_{j=1}^{i-1} L_{ij} x_j}{L_{ii}}, \quad i = 2, 3, \dots, n$$

$n$  divisions

$n(n-1)/2$  subtractions/additions

$n(n-1)/2$  multiplications



Computational complexity is  $O(n^2)$

# Linear System of Equations

How do we solve  $\mathbf{A} \mathbf{x} = \mathbf{b}$  when  $\mathbf{A}$  is a non-triangular matrix?

We can perform LU factorization: given a  $n \times n$  matrix  $\mathbf{A}$ , obtain lower triangular matrix  $\mathbf{L}$  and upper triangular matrix  $\mathbf{U}$  such that

$$\mathbf{A} = \mathbf{L}\mathbf{U}$$

where we set the diagonal entries of  $\mathbf{L}$  to be equal to 1.

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ L_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \dots & 1 \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1n} \\ 0 & U_{22} & \dots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & U_{nn} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix}$$

# LU Factorization

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ L_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \dots & 1 \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1n} \\ 0 & U_{22} & \dots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & U_{nn} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix}$$

Assuming the LU factorization is known, we can solve the general system



# Example

Assume the  $\mathbf{A} = \mathbf{LU}$  factorization is known, yielding:

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 1 & 1 & 0 \\ 0.5 & 0.5 & 0.5 & 1 \end{pmatrix} \quad \mathbf{U} = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & -2 & 1 & 2.5 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & 0.75 \end{pmatrix}$$

Determine the solution  $\mathbf{x}$  that satisfies  $\mathbf{Ax} = \mathbf{b}$ , when  $\mathbf{b} = \begin{pmatrix} 2 \\ 2 \\ 1 \\ 4 \end{pmatrix}$

$$\underbrace{\mathbf{L} \mathbf{U} \mathbf{x}}_{\mathbf{y}} = \mathbf{b}$$

First, solve the lower-triangular system  $\mathbf{L} \mathbf{y} = \mathbf{b}$  for the variable  $\mathbf{y}$

Then, solve the upper-triangular system  $\mathbf{U} \mathbf{x} = \mathbf{y}$  for the variable  $\mathbf{x}$

# 2x2 LU Factorization (simple example)

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ L_{21} & 1 \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix}$$

# Example

Which of the following statements about the  $A = LU$  factorization are true?

- 1)  It can happen that  $A$  is invertible, but  $U$  is not.
- 2)   $U$  has a diagonal that is full of ones.
- 3)  A factorization  $A = LU$  does not always exist.
- 4)   $L$  has a diagonal that is full of ones.

A) 1 and 2

B) 1 and 3

C) 2 and 3

D) 2 and 4

E) 3 and 4

# Example

## Computing the Lower-Triangular Factor in LU

1 point

Consider the matrix

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

and its corresponding LU factorization ( $A = LU$ ), where the lower and upper triangular matrices given respectively by

$$L = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}.$$

$l_{21} =$	<input type="text"/>
$u_{11} =$	<input type="text"/>
$u_{12} =$	<input type="text"/>
$u_{22} =$	<input type="text"/>

# LU Factorization

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix} =$$

# LU Factorization

$a_{11}$ : scalar

$\mathbf{a}_{12}$ : row vector ( $1 \times (n - 1)$ )

$\mathbf{a}_{21}$ : column vector ( $(n - 1) \times 1$ )

$\mathbf{A}_{22}$ : matrix ( $(n - 1) \times (n - 1)$ )

$$\begin{pmatrix}
 \boxed{A_{11}} & \boxed{A_{12} \quad \dots \quad A_{1n}} \\
 \boxed{A_{21}} & \boxed{A_{22} \quad \dots \quad A_{2n}} \\
 \vdots & \vdots \quad \ddots \quad \vdots \\
 \boxed{A_{n1}} & \boxed{A_{n2} \quad \dots \quad A_{nn}}
 \end{pmatrix}
 = \begin{pmatrix}
 a_{11} & \mathbf{a}_{12} \\
 \mathbf{a}_{21} & \mathbf{A}_{22}
 \end{pmatrix}
 = \begin{pmatrix}
 1 & \mathbf{0} \\
 \mathbf{l}_{21} & \mathbf{L}_{22}
 \end{pmatrix}
 \begin{pmatrix}
 \mathbf{u}_{11} & \mathbf{u}_{12} \\
 \mathbf{0} & \mathbf{U}_{22}
 \end{pmatrix}$$

1) First row of  $\mathbf{U}$  is the first row of  $\mathbf{A}$



$$\begin{pmatrix}
 \boxed{a_{11}} & \boxed{\mathbf{a}_{12}} \\
 \boxed{\mathbf{a}_{21}} & \boxed{\mathbf{A}_{22}}
 \end{pmatrix}
 = \begin{pmatrix}
 \boxed{u_{11}} & \boxed{\mathbf{u}_{12}} \\
 \boxed{u_{11} \mathbf{l}_{21}} & \boxed{\mathbf{l}_{21} \mathbf{u}_{12} + \mathbf{L}_{22} \mathbf{U}_{22}}
 \end{pmatrix}$$

2)  $\mathbf{l}_{21} = \frac{1}{u_{11}} \mathbf{a}_{21}$

First column of  $\mathbf{L}$  is the first column of  $\mathbf{A} / u_{11}$



3)  $\mathbf{M} = \mathbf{L}_{22} \mathbf{U}_{22} = \overbrace{\mathbf{A}_{22} - \mathbf{l}_{21} \mathbf{u}_{12}}^{\text{Known!}}$

Need another factorization!

# Example

$$M = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 6 & 2 \\ 1 & 3 & 4 & 2 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \end{pmatrix}$$

$$U = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & -2 & 1 & 2.5 \\ 1 & -2 & 4 & 1.5 \\ 1 & -1 & 2 & 1.5 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \end{pmatrix}$$

$$U = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & -2 & 1 & 2.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & -2 & 1 & 2.5 \\ 1 & -2 & 3 & -1 \\ 1 & -1 & 1.5 & 0.25 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 1 & 1 & 0 \\ 0.5 & 0.5 & 0.5 & 0 \end{pmatrix}$$

$$U = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & -2 & 1 & 2.5 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & -2 & 1 & 2.5 \\ 1 & -2 & 3 & -1 \\ 1 & -1 & 1.5 & 0.75 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 1 & 1 & 0 \\ 0.5 & 0.5 & 0.5 & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & -2 & 1 & 2.5 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & 0.75 \end{pmatrix}$$

# LU Algorithm

```
A = np.array([[2.0,8,4,1],[1,2,3,3],[1,2,6,2],[1,3,4,2]])
```

```
## Algorithm 1
## Factorization using the block-format,
## creating new matrices L and U
## and not modifying A
print("LU factorization using Algorithm 1")
L = np.zeros((n,n))
U = np.zeros((n,n))
M = A.copy()
for i in range(n):
    U[i,i:] = M[i,i:]
    L[i:,i] = M[i:,i]/U[i,i]
    M[i+1:,i+1:] -= np.outer(L[i+1:,i],U[i,i+1:])
```

```
## Algorithm 2
## Factorization using the block-format
## Matrices L and U are stored in the input matrix
## that could be a copy of A or A itself
print("LU factorization using Algorithm 2")
M = A.copy()
for i in range(n):
    M[i+1:,i] = M[i+1:,i]/M[i,i]
    M[i+1:,i+1:] -= np.outer(M[i+1:,i],M[i,i+1:])
```



# LU Algorithm

```
A = np.array([[2.0,8,4,1],[1,2,3,3],[1,2,6,2],[1,3,4,2]])
```

```
## Algorithm 2  
## Factorization using the block-format  
## Matrices L and U are stored in the input matrix  
## that could be a copy of A or A itself  
print("LU factorization using Algorithm 2")
```

```
M = A.copy()
```

```
for i in range(n):
```

```
    M[i+1:,i] = M[i+1:,i]/M[i,i]
```

```
    M[i+1:,i+1:] -= np.outer(M[i+1:,i],M[i,i+1:])
```

```
    For j = i + 1:n - 1
```

```
         $M[j,i] = M[j,i]/M[i,i]$ 
```

```
    For j = i + 1:n - 1
```

```
        For k = i + 1:n - 1
```

```
             $M[j,k] -= M[j,i] * M[i,k]$ 
```

```
M = A.copy()
```

```
for i in range(n-1):
```

```
    for j in range(i+1,n):
```

```
        M[j,i] = M[j,i]/M[i,i]
```

```
        for k in range(i+1,n):
```

```
            M[j,k] -= M[j,i]*M[i,k]
```

# Cost of LU factorization

```
## Algorithm 1
## Factorization using the block-format,
## creating new matrices L and U
## and not modifying A
print("LU factorization using Algorithm 1")
L = np.zeros((n,n))
U = np.zeros((n,n))
M = A.copy()
for i in range(n):
    U[i,i:] = M[i,i:]
    L[i:,i] = M[i:,i]/U[i,i]
    M[i+1:,i+1:] -= np.outer(L[i+1:,i],U[i,i+1:])
```

Side note:

$$\sum_{i=1}^m i = \frac{1}{2}m(m+1)$$

$$\sum_{i=1}^m i^2 = \frac{1}{6}m(m+1)(2m+1)$$

# Example

Which of the following statements are true about the LU factorization of an  $n \times n$  matrix  $A$ , assuming LU factorization of  $A$  exists and not considering any row/column interchanges?

**Select all that apply:**

- 1)  $A = LU$ .
- 2) LU factorization is exactly performing Gaussian elimination.
- 3) We can solve for  $LUx = b$  instead of solving  $Ax = b$  to obtain  $x$ .
- 4)  $L$  is a lower triangular matrix, and is exactly the lower part of  $A$  but with unit diagonal.
- 5)  $U$  is an upper triangular matrix, and is exactly the upper part of  $A$  (including diagonal).

A) 1, 2, 3

B) 1, 2, 3, 5

C) 1, 3

D) 1, 2, 3, 4, 5

E) 4, 5

# Solving linear systems

In general, we can solve a linear system of equations following the steps:

- 1) Factorize the matrix  $\mathbf{A}$  :  $\mathbf{A} = \mathbf{LU}$  (complexity  $O(n^3)$ )
- 2) Solve  $\mathbf{L}\mathbf{y} = \mathbf{b}$  (complexity  $O(n^2)$ )
- 3) Solve  $\mathbf{U}\mathbf{x} = \mathbf{y}$  (complexity  $O(n^2)$ )

But why should we decouple the factorization from the actual solve?  
(Remember from Linear Algebra, Gaussian Elimination does not decouple these two steps...)

# Example

Let's assume that when solving the system of equations  $\mathbf{K} \mathbf{U} = \mathbf{F}$ , we observe the following:

- When the matrix  $\mathbf{K}$  has dimensions (100,100), computing the LU factorization takes about 1 second and each solve (forward + backward substitution) takes about 0.01 seconds.

Estimate the total time it will take to find the response  $\mathbf{U}$  corresponding to 10 different vectors  $\mathbf{F}$  when the matrix  $\mathbf{K}$  has dimensions (1000,1000)?

- A)  $\sim 10$  seconds
- B)  $\sim 10^2$  seconds
- C)  $\sim 10^3$  seconds
- D)  $\sim 10^4$  seconds
- E)  $\sim 10^5$  seconds

# What can go wrong with the previous algorithm?

Demo "Little c"

$$\mathbf{M} = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & 4 & 3 & 3 \\ 1 & 2 & 6 & 2 \\ 1 & 3 & 4 & 2 \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{U} = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$l_{21}\mathbf{u}_{12} = \begin{pmatrix} 4 & 2 & 0.5 \\ 4 & 2 & 0.5 \\ 4 & 2 & 0.5 \end{pmatrix}$$

$$\mathbf{M} - l_{21}\mathbf{u}_{12} = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & 0 & 1 & 2.5 \\ 1 & -2 & 4 & 1.5 \\ 1 & -1 & 2 & 1.5 \end{pmatrix}$$

The next update for the lower triangular matrix will result in a division by zero! LU factorization fails.

What can we do to get something like an LU factorization?

# Pivoting

Approach:

1. Swap rows if there is a zero entry in the diagonal
2. Even better idea: Find the largest entry (by absolute value) and swap it to the top row.

The entry we divide by is called the pivot.

Swapping rows to get a bigger pivot is called (partial) pivoting.

$$\begin{pmatrix} a_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & \mathbf{A}_{22} \end{pmatrix} = \begin{pmatrix} u_{11} & \mathbf{u}_{12} \\ u_{11} l_{21} & l_{21} \mathbf{u}_{12} + \mathbf{L}_{22} \mathbf{U}_{22} \end{pmatrix}$$



Find the largest entry (in magnitude)

# LU Factorization with Partial Pivoting



# Example

$$A = M = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 2 \\ 1 & 3 & 4 & 2 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \end{pmatrix} \quad U = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & -2 & 1 & 2.5 \\ 1 & -2 & 1 & 1.5 \\ 1 & -1 & 2 & 1.5 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \end{pmatrix} \quad U = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & -2 & 1 & 2.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & -2 & 1 & 2.5 \\ 1 & -2 & 0 & -1 \\ 1 & -1 & 1.5 & 0.25 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0.5 & 1.0 & 0 & 0 \end{pmatrix} \quad U = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & -2 & 1 & 2.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 1 & -2 & 1 & 2.5 \\ 1 & -1 & 1.5 & 0.25 \\ 1 & -2 & 0 & -1 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 0.5 & 1.0 & 0 \\ 0.5 & 1.0 & 0 & 1.0 \end{pmatrix} \quad U = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & -2 & 1 & 2.5 \\ 0 & 0 & 1.5 & 0.25 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

# Demo “Pivoting example”

$$A = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix}$$

$$\bar{A} = PA = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix} = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 \\ 0.75 & 0 & 0 & 0 \end{pmatrix} \quad U = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad l_{21}\mathbf{u}_{12} = \begin{pmatrix} 3.5 & 4.5 & 2.5 \\ 1.75 & 2.25 & 1.25 \\ 5.25 & 6.75 & 3.75 \end{pmatrix}$$

$$\bar{A} - l_{21}\mathbf{u}_{12} = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 4 & -0.5 & -1.5 & -1.5 \\ 2 & -0.75 & -1.25 & -1.25 \\ 6 & 1.75 & 2.25 & 4.25 \end{pmatrix}$$

# Demo “Pivoting example”

$$\bar{A} = \bar{A} - l_{21} \mathbf{u}_{12} = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 4 & -0.5 & -1.5 & -1.5 \\ 2 & -0.75 & -1.25 & -1.25 \\ 6 & 1.75 & 2.25 & 4.25 \end{pmatrix}$$

$$\bar{A} = P\bar{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 8 & 7 & 9 & 5 \\ 6 & 1.75 & 2.25 & 4.25 \\ 2 & -0.75 & -1.25 & -1.25 \\ 4 & -0.5 & -1.5 & -1.5 \end{pmatrix} = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 6 & 1.75 & 2.25 & 4.25 \\ 2 & -0.75 & -1.25 & -1.25 \\ 4 & -0.5 & -1.5 & -1.5 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 1 & 0 & 0 \\ 0.25 & -0.428 & 0 & 0 \\ 0.5 & -0.285 & 0 & 0 \end{pmatrix} \quad U = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 0 & 1.75 & 2.25 & 4.25 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad l_{21} \mathbf{u}_{12} = \begin{pmatrix} -0.963 & -1.819 \\ -0.6412 & -1.2112 \end{pmatrix}$$

$$\bar{A} = \bar{A} - l_{21} \mathbf{u}_{12} = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 6 & 1.75 & 2.25 & 4.25 \\ 2 & -0.75 & -0.287 & 0.569 \\ 4 & -0.5 & -0.8587 & -0.2887 \end{pmatrix}$$

# Demo “Pivoting example”

$$\bar{A} = \bar{A} - l_{21}u_{12} = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 6 & 1.75 & 2.25 & 4.25 \\ 2 & -0.75 & -0.287 & 0.569 \\ 4 & -0.5 & -0.8587 & -0.2887 \end{pmatrix}$$

$$\bar{A} = P\bar{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 8 & 7 & 9 & 5 \\ 6 & 1.75 & 2.25 & 4.25 \\ 2 & -0.75 & -0.287 & 0.569 \\ 4 & -0.5 & -0.8587 & -0.2887 \end{pmatrix} = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 6 & 1.75 & 2.25 & 4.25 \\ 4 & -0.5 & -0.8587 & -0.2887 \\ 2 & -0.75 & -0.287 & 0.569 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 1 & 0 & 0 \\ 0.5 & -0.285 & 1 & 0 \\ 0.25 & -0.428 & 0.334 & 0 \end{pmatrix} \quad U = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 0 & 1.75 & 2.25 & 4.25 \\ 0 & 0 & -0.86 & -0.29 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 1 & 0 & 0 \\ 0.5 & -0.285 & 1 & 0 \\ 0.25 & -0.428 & 0.334 & 1 \end{pmatrix} \quad U = \begin{pmatrix} 8 & 7 & 9 & 5 \\ 0 & 1.75 & 2.25 & 4.25 \\ 0 & 0 & -0.86 & -0.29 \\ 0 & 0 & 0 & 0.67 \end{pmatrix}$$

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$