Solving Linear System of Equations

The "Undo" button for Linear Operations

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

$$\boldsymbol{y} = \boldsymbol{A}\,\boldsymbol{x}$$

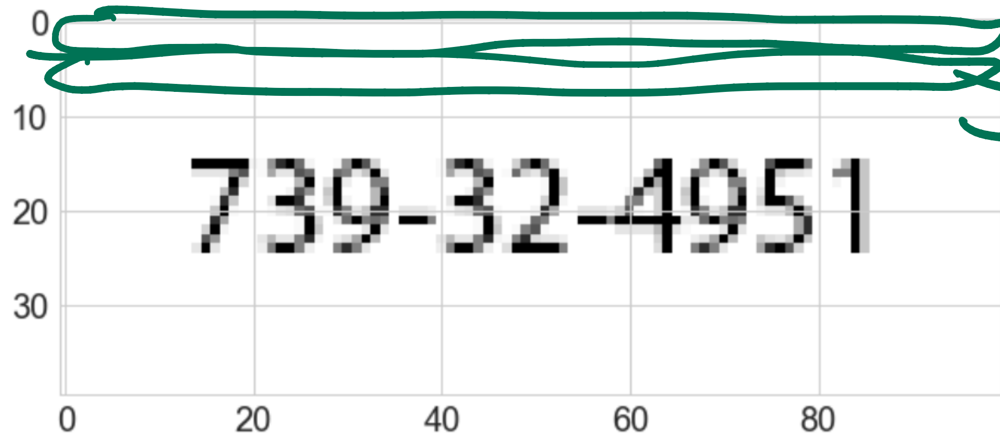
$$\boldsymbol{x} \xrightarrow[\text{transformation}]{\boldsymbol{A}} \boldsymbol{y}$$

What if we know $\boldsymbol{y}$ but not $\boldsymbol{x}$? How can we "undo" the transformation?

$$\boldsymbol{y} \xrightarrow[?]{\boldsymbol{A}^{-1}} \boldsymbol{x}$$

Solve $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$

Image Blurring Example

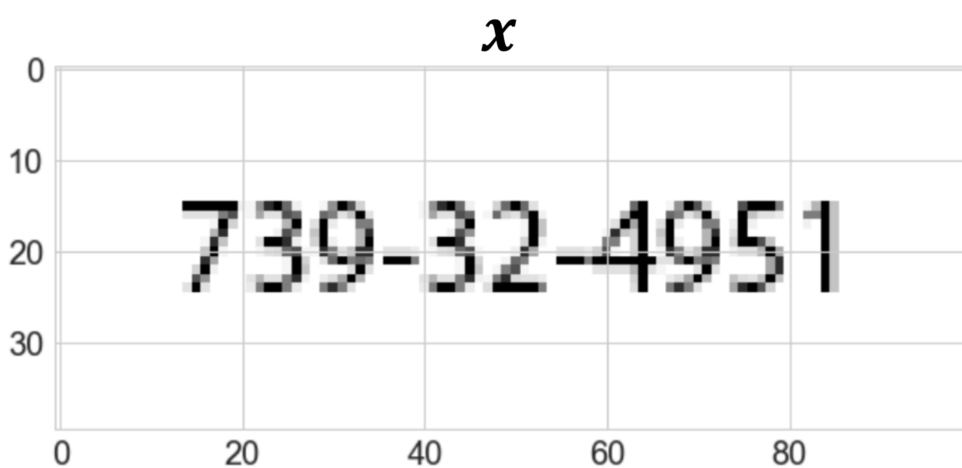


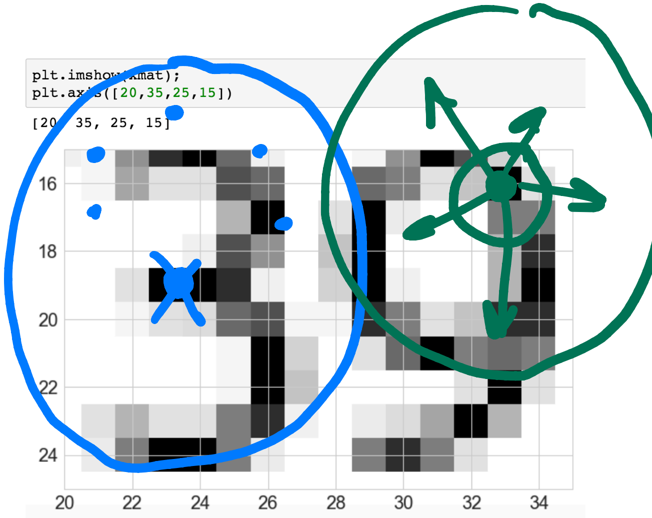
- Image is stored as a 2D array of real numbers between 0 and 1 (0 represents a white pixel, 1 represents a black pixel)
- ***xmat*** has 40 rows of pixels and 100 columns of pixels (40, 100)
- Flatten the 2D array as a 1D array
- $\boldsymbol{x}$ contains the 1D data with dimension 4000,
- Apply blurring operation to data $\boldsymbol{x}$, i.e.

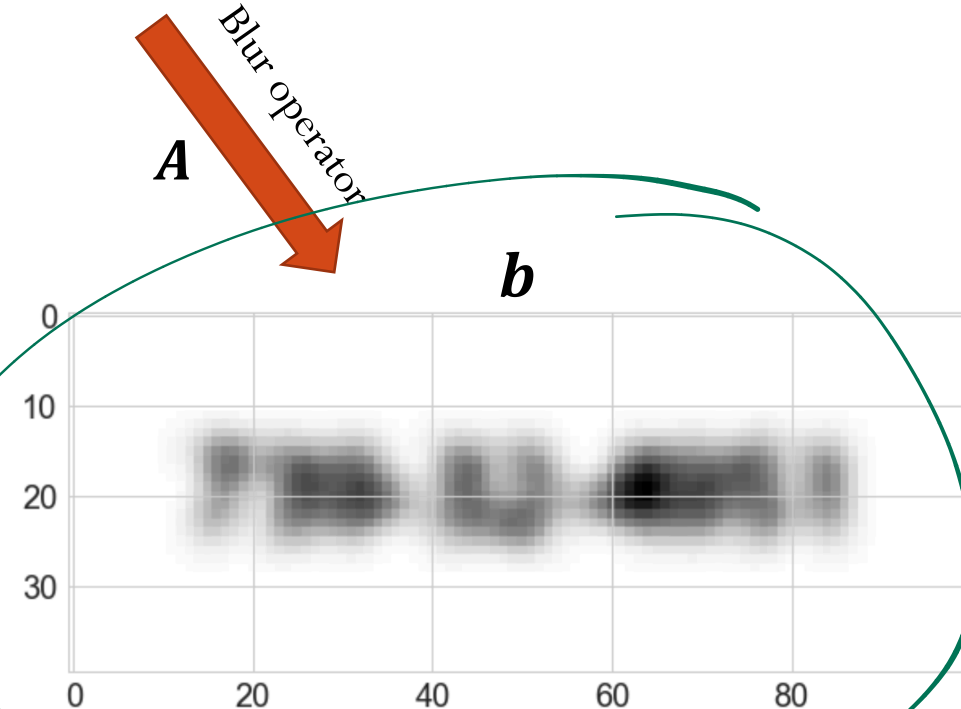
$$\boldsymbol{b} = \boldsymbol{A}\,\boldsymbol{x}$$

where $\boldsymbol{A}$ is the blur operator and $\boldsymbol{b}$ is the blurred image

Blur operator





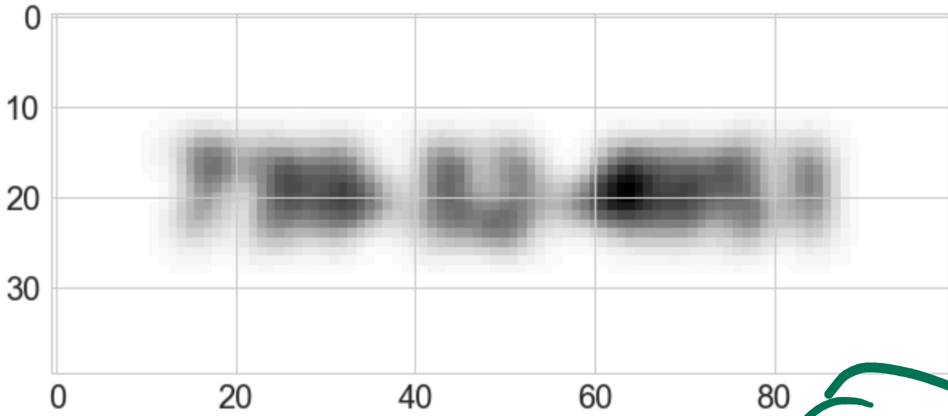


blurred image (4000,)	Blur operator (4000,4000)	"original" image (4000,)

$$\boldsymbol{b} = \boldsymbol{A}\,\boldsymbol{x}$$

"Undo" Blur to recover original image

$\boldsymbol{b}$

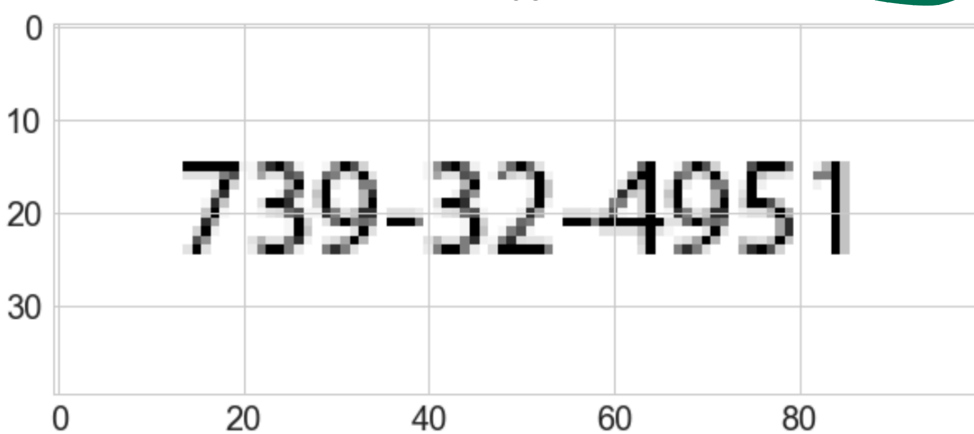


Solve

$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$

for $\boldsymbol{x}$

$\boldsymbol{x}$



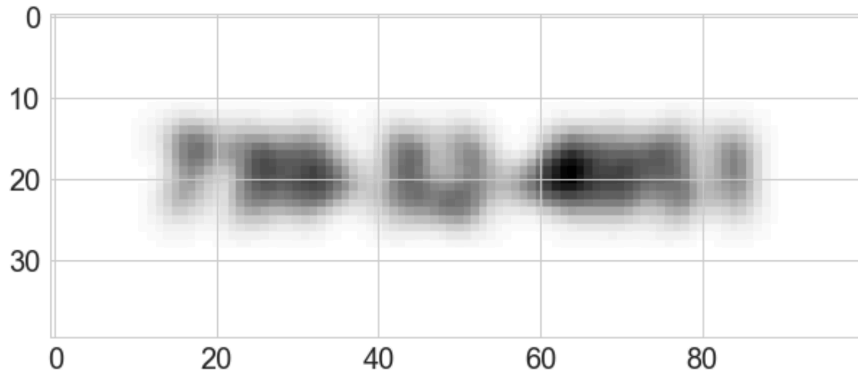
Assumptions:

1. we know the blur operator $\boldsymbol{A}$
2. the data set $\boldsymbol{b}$ does not have any noise ("clean data"

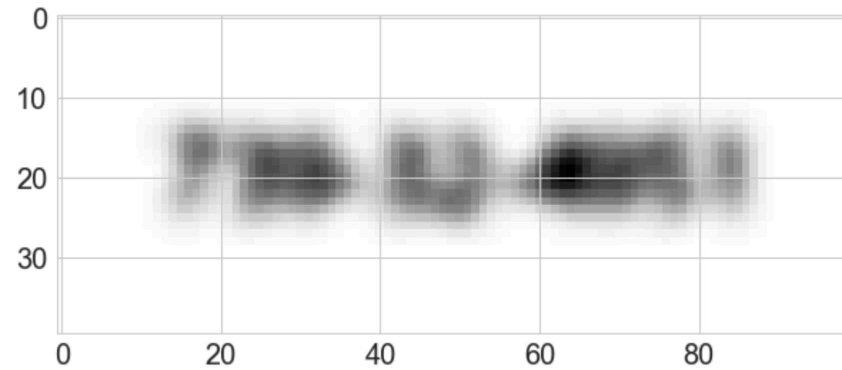
What happens if we add some noise to $\boldsymbol{b}$?

"Undo" Blur to recover original image

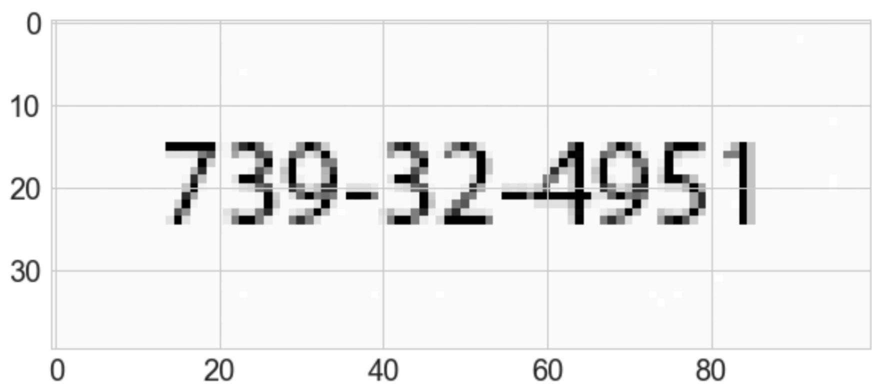
$\boldsymbol{y} + a * 10^{-6}\ (a \in (0,1))$

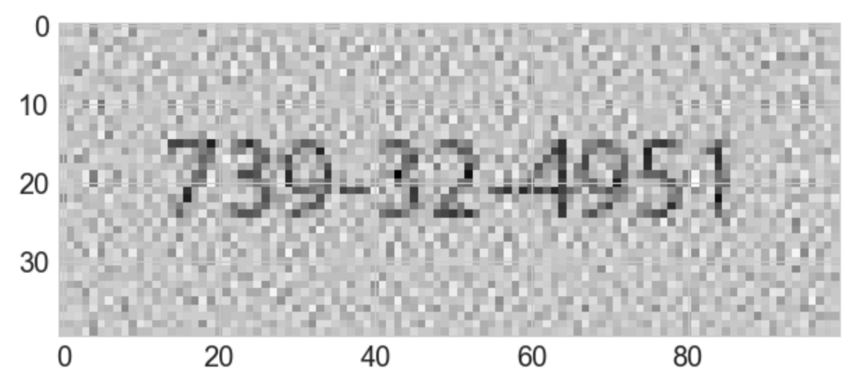


$\boldsymbol{y} + a * 10^{-4}\ (a \in (0,1))$



Solve $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$





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 $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$?

We can start with an "easier" system of equations…

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

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

Forward

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

Backward

Example: Forward-substitution for lower triangular systems

L_{11}

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

$$x_1 = b_1 / L_{11}$$

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} L_{ij} x_j}{L_{ii}} \qquad i = 2, 3, \ldots, n$$

$$2\,x_1 = 2 \rightarrow x_1 = 1$$

$$3\,x_1 + 2\,x_2 = 2 \rightarrow x_2 = \frac{2-3}{2} = -0.5$$

$$1\,\cancel{x_1} + 2\,\cancel{x_2} + 6\,x_3 = 6 \rightarrow x_3 = \frac{6-1+1}{6} = 1.0$$

$$1\,\cancel{x_1} + 3\,\cancel{x_2} + 4\,\cancel{x_3} + 2\,x_4 = 4 \rightarrow x_3 = \frac{4-1+1.5-4}{2} = 0.25$$

$$\boxed{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ -0.5 \\ 1.0 \\ 0.25 \end{pmatrix}}$$

Example: Backward-substitution for upper triangular systems

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

$$x_n = b_n / U_{nn}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} U_{ij} x_j}{U_{ii}}$$

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

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

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

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

LU Factorization

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

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

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

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

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

LU Factorization

$LU = A$

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

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

1. $LU = A \longrightarrow Ax = b$

$$L\underbrace{Ux}_{y} = b$$

2. $Ly = b \longrightarrow$ Forward solve for y

3. $Ux = y \longrightarrow$ Backward solve for $\underline{\underline{x}}$

LU Factorization (with pivoting)

$Ax = b$

Factorize: $\boxed{A = PLU}$

$\rightarrow \underbrace{PLU\,x}_{y} = b$

$P \rightarrow$ orthogonal

L

$P^T = P^{-1}$

$PLy = b \Rightarrow Ly = P^{-1}b$

Forward-substitution $\quad L\,y = P^T b \quad$ (Solve for $\boldsymbol{y}$)

Backward-substitution $\quad U\,x = y \quad$ (Solve for $\boldsymbol{x}$)

Example

Assume the $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ factorization is known, yielding:

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

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

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

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

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 1 & 1 & 0 \\ 0.5 & 0.5 & 0.5 & 1 \end{pmatrix} \boldsymbol{y} = \begin{pmatrix} 2 \\ 2 \\ 1 \\ 4 \end{pmatrix}$$

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

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

Methods to solve linear system of equations

$$\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$$

- LU
- Cholesky ① $A = LL^T \rightarrow \underbrace{LL^T x}_{y} = b$

symm pos.def ② $Ly = b$ ③ $L^T x = y$

- Sparse

$$\boxed{Ax = b}$$

LU Factorization - Algorithm

1. $A = LU$
2. $Ly = b$
3. $Ux = y$

2x2 LU Factorization (simple example)

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

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

$$A_{21} = L_{21}U_{11} \rightarrow \boxed{L_{21} = \frac{A_{21}}{U_{11}} = \frac{A_{21}}{A_{11}}}$$

$$A_{22} = L_{21}U_{12} + U_{22} \rightarrow \boxed{U_{22} = A_{22} - L_{21}U_{12}}$$

LU Factorization

a_{11}: scalar
$\boldsymbol{a}_{12}$: row vector $(1\times(n-1))$
$\boldsymbol{a}_{21}$: column vector $((n-1)\times 1)$
$\boldsymbol{A}_{22}$: matrix $((n-1)\times(n-1))$

$A_{n\times n}$

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

a_{11}, $\boldsymbol{a}_{12}$, $\boldsymbol{a}_{21}$, $\boldsymbol{A}_{22}$

row

col

$u_{12} = a_{12}$

$u_{11} = a_{11}$

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

column vector

col

$l_{21} = \dfrac{a_{21}}{a_{11}}$ ← scalar

$A_{22} = l_{21} u_{12} + L_{22} U_{22}$

$L_{22} U_{22} = A_{22} - l_{21} u_{12}$

$L_{22} = ?$

$U_{22} = ?$

recursion

LU Factorization

a_{11}: scalar
$\boldsymbol{a}_{12}$: row vector $(1\times(n-1))$
$\boldsymbol{a}_{21}$: column vector $((n-1)\times 1)$
$\boldsymbol{A}_{22}$: matrix $((n-1)\times(n-1))$

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

a_{11} $\boldsymbol{a}_{12}$ $\boldsymbol{a}_{21}$ $\boldsymbol{A}_{22}$

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

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

2) $\boldsymbol{l}_{21} = \dfrac{1}{u_{11}}\boldsymbol{a}_{21}$

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

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

Need another factorization!

Example

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

1) First row of $\boldsymbol{U}$ is the first row of $\boldsymbol{A}$
2) First column of $\boldsymbol{L}$ is the first column of $\boldsymbol{A}/\ u_{11}$
3) $\boldsymbol{L}_{22}\boldsymbol{U}_{22} = \boldsymbol{A}_{22} - \underline{\boldsymbol{l}_{21}\boldsymbol{u}_{12}}$

u_{12}

$$\underset{=}{\boldsymbol{L}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \end{pmatrix} \qquad \underset{=}{\boldsymbol{U}} = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

l_{21}

$$\boldsymbol{L}_{22}\boldsymbol{U}_{22} = \boldsymbol{A}_{22} - \boldsymbol{l}_{21}\boldsymbol{u}_{12} = \underline{\begin{pmatrix} 2 & 3 & 3 \\ 2 & 6 & 2 \\ 3 & 4 & 2 \end{pmatrix}} - \underline{\begin{pmatrix} 4 & 2 & 0.5 \\ 4 & 2 & 0.5 \\ 4 & 2 & 0.5 \end{pmatrix}}$$

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

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

$$\frac{-2}{-2} = 1 \qquad -\frac{1}{-2} = \frac{1}{2}$$

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \end{pmatrix} \qquad \boldsymbol{U} = \begin{pmatrix} 2 & 8 & 4 & 1 \\ 0 & -2 & 1 & 2.5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{L}_{22}\boldsymbol{U}_{22} = \boldsymbol{A}_{22} - \boldsymbol{l}_{21}\boldsymbol{u}_{12} = \begin{pmatrix} 4 & 1.5 \\ 2 & 1.5 \end{pmatrix} - \begin{pmatrix} 1 & 2.5 \\ 0.5 & 1.25 \end{pmatrix}$$

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

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

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

$$
\boldsymbol{L}_{22}\boldsymbol{U}_{22} = \boldsymbol{A}_{22} - \boldsymbol{l}_{21}\boldsymbol{u}_{12} = \boxed{0.25} - (-0.5) = 0.75
$$

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

LU Factorization

a_{11}: scalar
$\boldsymbol{a}_{12}$: row vector $(1\times(n-1))$
$\boldsymbol{a}_{21}$: column vector $((n-1)\times 1)$
$\boldsymbol{A}_{22}$: matrix $((n-1)\times(n-1))$

a_{11} $\boldsymbol{a}_{12}$

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

$\boldsymbol{a}_{21}$ $\boldsymbol{A}_{22}$

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

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

2) $\boldsymbol{l}_{21} = \dfrac{1}{u_{11}}\boldsymbol{a}_{21}$

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

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

Need another factorization!

Cost of solving linear system of equations

Cost of solving triangular systems

$$x_{n-1} = \frac{b_{n-1} - U_{n-1,n}\,x_n}{U_{n-1,n-1}}$$

$$x_n = b_n/U_{nn}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} U_{ij}x_j}{U_{ii}}, \qquad i = n-1, n-2, \dots, 1$$

	$i=n$	$i=n-1$	$i=n-2$	...	$i=n-(n-1)$ $=1$	TOTAL
#divisions	1	1	1	...	1	n
#multiplications	0	1	2	3 ...	$(n-1)$	$\frac{1}{2}n(n-1)$
#subtractions + additions	0	1	2	3 ...	$(n-1)$	$\frac{1}{2}n(n-1)$

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

$+$

Total $= n + n(n-1)$

Comp. complexity $n \to \infty$ $O(n^2)$

Cost of solving triangular systems

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

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

→ Computational complexity is $O(n^2)$

$n \to \infty$

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

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

→ Computational complexity is $O(n^2)$

Cost of LU factorization

Side note:

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

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

```
## 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:])
```

$(n-1)$

$(n-1)$

$(n-1)^2$

$(n-2)^2$

divi $(n-1)+(n-2)+(n-3)+\cdots+1 = n(n-1)/2$

mult. $(n-1)^2+(n-2)^2+\cdots+1 = \frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6}$

sub/ad: $(n-1)^2+(n-2)^2+\cdots+1 = \frac{n^3}{3} - \frac{n}{2} + \frac{n}{6}$

TOTAL $\quad n \to \infty \quad \frac{2n^3}{3} \longrightarrow O(n^3)$

Solving linear systems

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

1) Factorize the matrix $\boldsymbol{A}$: $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ (complexity $O(n^3)$)

2) Solve $\boldsymbol{L}\,\boldsymbol{y} = \boldsymbol{b}$ (complexity $O(n^2)$)

3) Solve $\boldsymbol{U}\,\boldsymbol{x} = \boldsymbol{y}$ (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…)

(1) LU factorization : $\sim \frac{2n^3}{3}$ $O(n^3)$ $n \to \infty$

(2) Cholesky : Factorization $\sim \frac{n^3}{3} \to O(n^3)$

(3) Matrix-matrix multiplication $\sim 2n^3 \to O(n^3)$

Example

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

- When the matrix $\boldsymbol{K}$ 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 $\boldsymbol{U}$ corresponding to 10 different vectors $\boldsymbol{F}$ when the matrix $\boldsymbol{K}$ has dimensions (1000,1000)?

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