

Numerical Computing & Images Tutorial

EECS 442

Fall 2024, University of Michigan

Colab Notebook Format

- Like jupyter notebooks on the web
- Execute cells of code
- Useful for immediate visualization (a lot of what we do!)
- Allows us to use GPUs remotely (important for ML)

Numpy Tutorial

Colab:

https://colab.research.google.com/drive/1f7nAcXVy7jgvt21LhP_dHSmM-ARqUXCD?usp=sharing

Image Loading & Manipulation

Colab:

<https://colab.research.google.com/drive/1uNOh4DIBRb3gA3rH32g3sCyGrcn2crip>

Problem Set 1

- You will work in Colab
- Examples from both of these notebooks will be helpful!

Linear Algebra Review

Vectors

- A vector $x \in \mathbb{R}^n$ is a stack of n real values.

$$\cdot x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = [x_1 \ x_2 \ \dots \ x_n]^T$$

- In the computer vision context, x can represent an image vector where each x_i is a pixel value in that image

Measuring Length - Norms

- Norms are a measure of the magnitude/ length of the vector
- Formally, a norm $\|\cdot\| : \mathbb{R}^n \rightarrow [0, \infty)$ is a non-negative valued function.

- Examples:

- Euclidean (l_2) norm (Default choice): $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$
- Manhattan distance (l_1 norm): $\|\mathbf{x}\|_1 = |x_1| + |x_2| + \dots + |x_n|$
- In general l_p norm ($p \geq 1$): $\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \dots + |x_n|^p)^{1/p}$
- l_∞ norm: $\|\mathbf{x}\|_\infty := \lim_{p \rightarrow \infty} \|\mathbf{x}\|_p = \max\{|x_1|, |x_2|, \dots, |x_n|\}$

$$\mathbf{x} = [2, 3]$$

Measuring Length

Magnitude / length / (L2) norm of vector

$$\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \left(\sum_i^n x_i^2 \right)^{1/2}$$

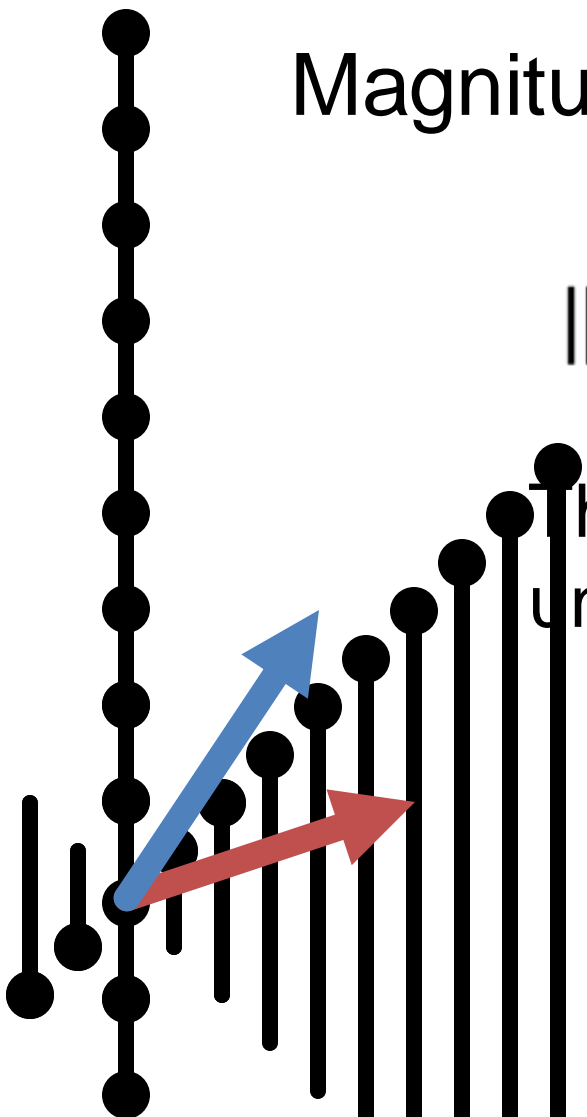
There are other norms; assume L2 unless told otherwise

$$\mathbf{y} = [3, 1]$$

$$\|\mathbf{x}\|_2 = \sqrt{13}$$

$$\|\mathbf{y}\|_2 = \sqrt{10}$$

Why?



Normalizing a Vector

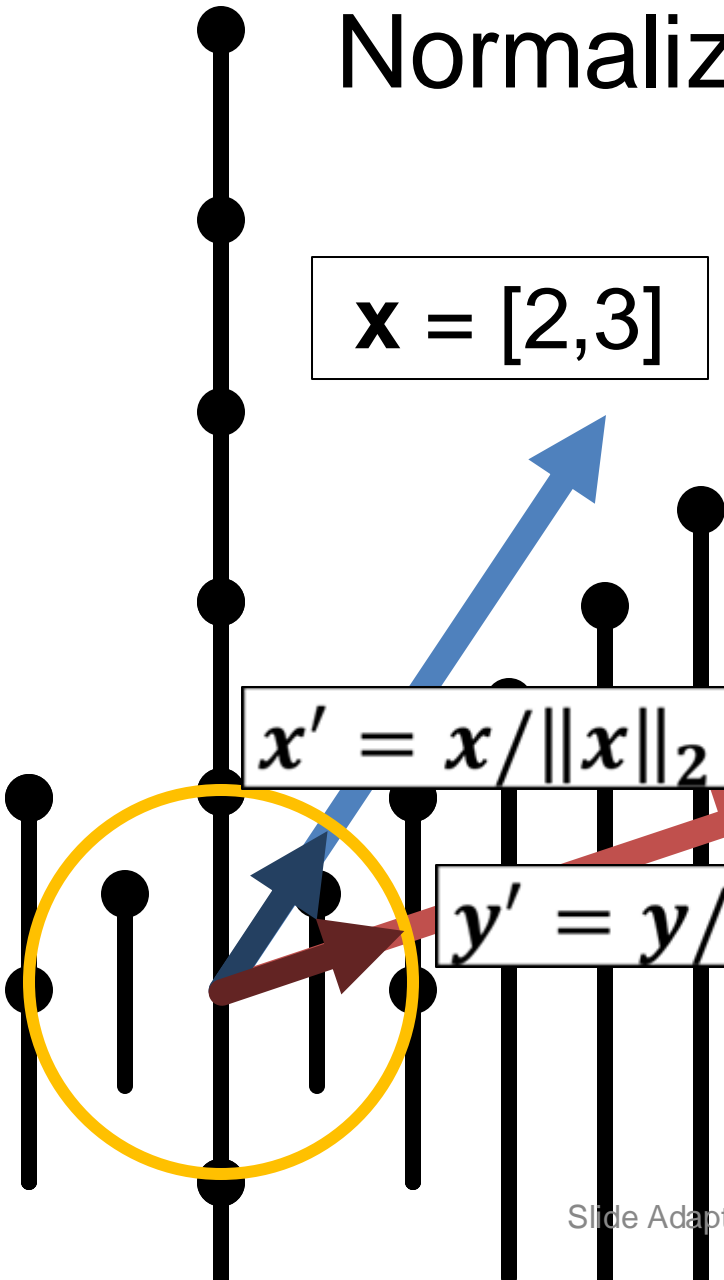
$$\mathbf{x} = [2, 3]$$

Dividing by norm gives something on the *unit sphere* (all vectors with length 1)

$$\mathbf{x}' = \mathbf{x} / \|\mathbf{x}\|_2$$

$$\mathbf{y} = [3, 1]$$

$$\mathbf{y}' = \mathbf{y} / \|\mathbf{y}\|_2$$

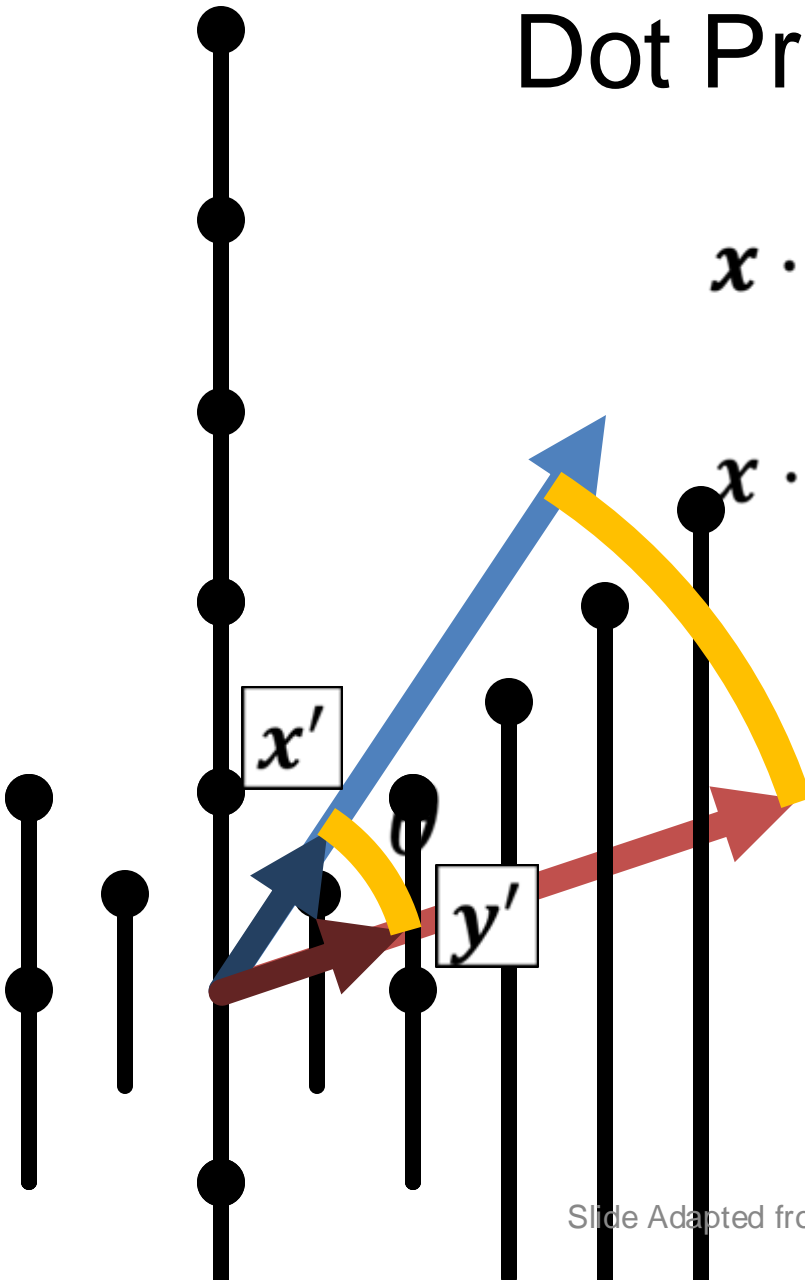


Dot Products

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^n x_i y_i = \mathbf{y}^T \mathbf{x}$$

$$\mathbf{x} \cdot \mathbf{y} = \cos(\theta) \|\mathbf{x}\| \|\mathbf{y}\|$$

What happens with
normalized / unit
vectors?



Dot Products

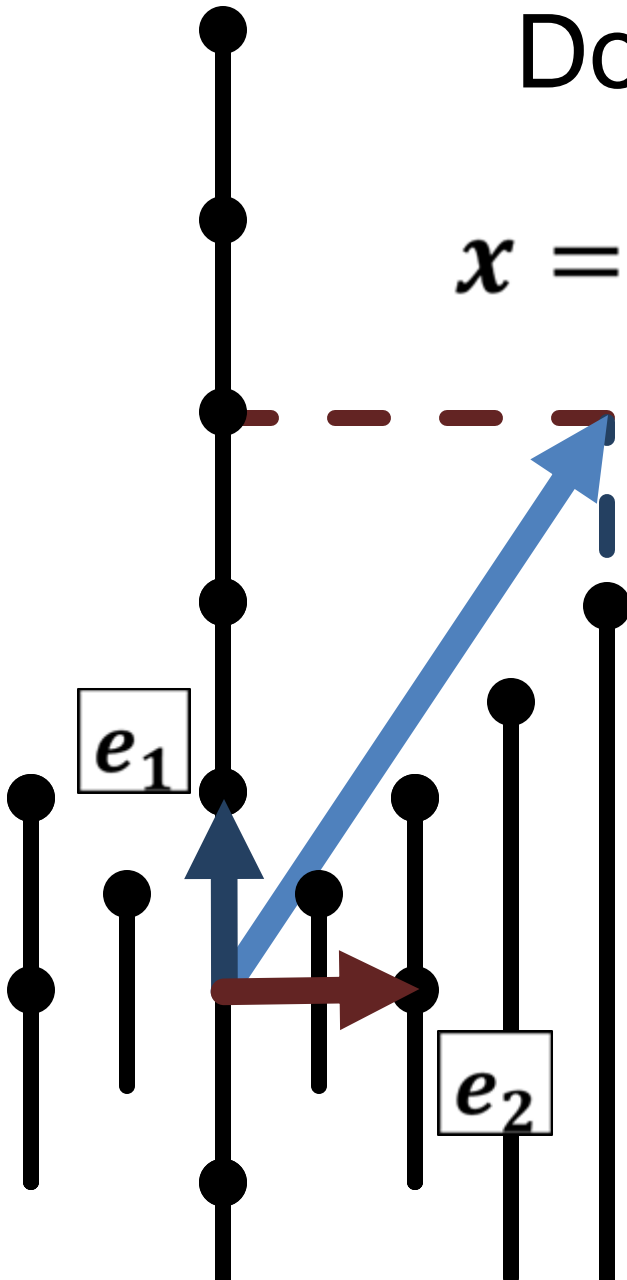
$$e_1 = [1,0]$$
$$e_2 = [0,1]$$

$$\mathbf{x} = [2,3] \quad \mathbf{x} \cdot \mathbf{y} = \sum_i^n x_i y_i$$

What's $\mathbf{x} \cdot \mathbf{e}_1$, $\mathbf{x} \cdot \mathbf{e}_2$?

Ans: $\mathbf{x} \cdot \mathbf{e}_1 = 2$; $\mathbf{x} \cdot \mathbf{e}_2 = 3$

- Dot product is projection
- Amount of \mathbf{x} that's also pointing in direction of \mathbf{y}



Dot Products

$$\mathbf{x} = [2,3] \quad \mathbf{x} \cdot \mathbf{y} = \sum_i^n x_i y_i$$

What's $\mathbf{x} \cdot \mathbf{x}$?

Ans: $\mathbf{x} \cdot \mathbf{x} = \sum x_i x_i = \|\mathbf{x}\|_2^2$

Matrices

Horizontally concatenate n , m -dim column vectors and you get a $m \times n$ matrix A (here 2×3)

$$\mathbf{A} = \left[\begin{array}{c|c|c} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ | & & | \end{array} \right] = \left[\begin{array}{|c|} \mathbf{v}_{1_1} \\ \mathbf{v}_{1_2} \end{array} \quad \begin{array}{|c|} \mathbf{v}_{2_1} \\ \mathbf{v}_{2_2} \end{array} \quad \begin{array}{|c|} \mathbf{v}_{3_1} \\ \mathbf{v}_{3_2} \end{array} \right]$$

Matrices

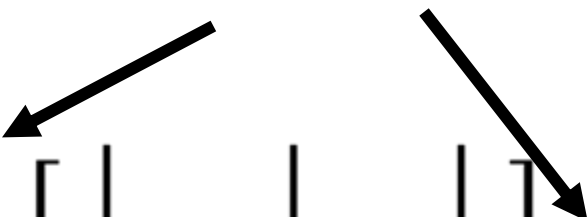
Transpose: flip rows / columns $\begin{bmatrix} a \\ b \\ c \end{bmatrix}^T = [a \quad b \quad c] \quad (3 \times 1)^T = 1 \times 3$

Vertically concatenate m , n -dim row vectors and you get a $m \times n$ matrix A (here 2×3)

$$A = \begin{bmatrix} - & \mathbf{u}_1^T & - \\ & \vdots & \\ - & \mathbf{u}_m^T & - \end{bmatrix} = \begin{bmatrix} u_{1_1} & u_{1_2} & u_{1_3} \\ u_{2_1} & u_{2_2} & u_{2_3} \end{bmatrix}$$

Matrix-Vector Product

$$\mathbf{y}_{2 \times 1} = \mathbf{A}_{2 \times 3} \mathbf{x}_{3 \times 1}$$


$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\mathbf{y} = x_1 \mathbf{v}_1 + x_2 \mathbf{v}_2 + x_3 \mathbf{v}_3$$

Linear combination of columns of \mathbf{A}

Matrix-Vector Product

$$\mathbf{y}_{2 \times 1} = \mathbf{A}_{2 \times 3} \mathbf{x}_{3 \times 1}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} - & \mathbf{u}_1^T & - \\ - & \mathbf{u}_2^T & - \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$y_1 = \mathbf{u}_1^T \mathbf{x} \quad y_2 = \mathbf{u}_2^T \mathbf{x}$$

Dot product between rows of \mathbf{A} and \mathbf{x}

Matrix Multiplication

Generally: \mathbf{A}_{mn} and \mathbf{B}_{np} yield product $(\mathbf{AB})_{mp}$

$$\mathbf{AB} = \begin{bmatrix} - & \mathbf{a}_1^T & - \\ & \vdots & \\ - & \mathbf{a}_m^T & - \end{bmatrix} \begin{bmatrix} | & & | \\ \mathbf{b}_1 & \cdots & \mathbf{b}_p \\ | & & | \end{bmatrix}$$

Yes – in \mathbf{A} , I'm referring to the rows, and in \mathbf{B} , I'm referring to the columns

Matrix Multiplication

Generally: \mathbf{A}_{mn} and \mathbf{B}_{np} yield product $(\mathbf{AB})_{mp}$

$$\mathbf{AB} = \begin{bmatrix} \text{---} a_1^T \text{---} \\ \vdots \\ \text{---} a_m^T \text{---} \end{bmatrix} \begin{bmatrix} \downarrow b_1 & \dots & \downarrow b_p \\ a_1^T b_1 & \dots & a_1^T b_p \\ \vdots & \ddots & \vdots \\ a_m^T b_1 & \dots & a_m^T b_p \end{bmatrix}$$

$$\mathbf{AB}_{ij} = a_i^T b_j$$

Slide Adapted from David Fouhey

Matrix Multiplication

- Inner Dimensions must match
- Product gets the outer dimension
- (Yes, it's associative): $ABx = (A)(Bx) = (AB)x$
- (*No it's not commutative*): $ABx \neq (BA)x \neq (BxA)$

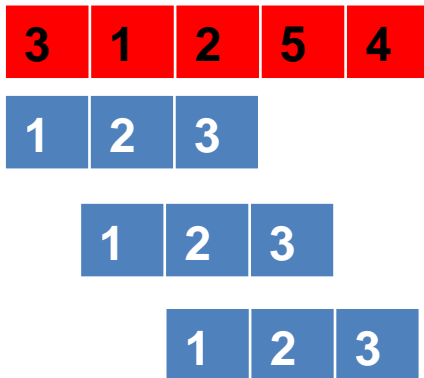
Cross-correlation

Consider 1D case for simplicity

- Correlation $c[m] = h * g = \sum_k h[m+k]g[k]$
- Convolution $f[m] = h \circ g = \sum_k h[m-k]g[k]$

Let $h = [3, 1, 2, 5, 4]$, $g = [1, 2, 3]$, then $c = [11, 20, 24]$:

$$c[0] = \sum_k h[0+k]g[k] = h[0]g[0] + h[1]g[1] + h[2]g[2] = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$



Each output element is from a dot product!

$$c[0] = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 3 + 2 + 6 = 11$$

$$c[1] = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 1 + 4 + 15 = 20$$

$$c[2] = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 2 + 10 + 12 = 24$$

Convolution

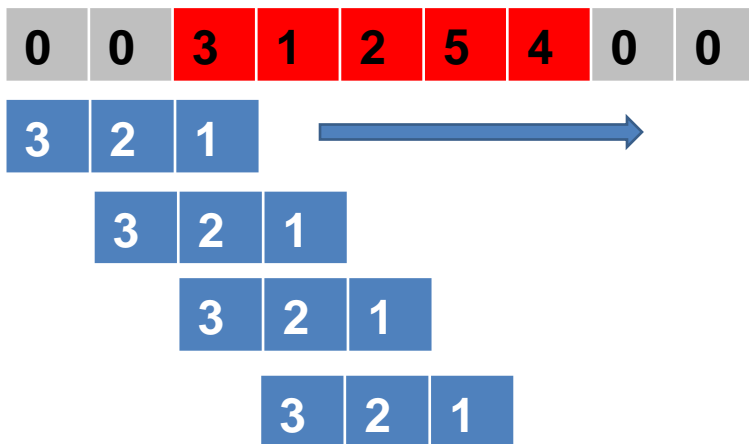
Consider 1D case for simplicity

- Correlation $c[m] = h * g = \sum_k h[m+k]g[k]$
- Convolution $f[m] = h \circ g = \sum_k h[m-k]g[k]$

Let $h = [3, 1, 2, 5, 4]$, $g = [1, 2, 3]$, then $f = [3, 7, 13, 12, 20, 23, 12]$:

$$f[0] = \sum_k h[0-k]g[k] = h[0]g[0] + h[-1]g[1] + h[-2]g[2] = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

Each output element is from a dot product!



3
7
13
12
20
23
12

Operations They Don't Teach

You Probably Saw Matrix Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a + e & b + f \\ c + g & d + h \end{bmatrix}$$

What is this? FYI: e is a scalar

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + e = \begin{bmatrix} a + e & b + e \\ c + e & d + e \end{bmatrix}$$

Broadcasting

If you want to be pedantic and proper, you expand e by multiplying a matrix of 1s (denoted $\mathbf{1}$)

$$\begin{aligned} \begin{bmatrix} a & b \\ c & d \end{bmatrix} + e &= \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \mathbf{1}_{2 \times 2} e \\ &= \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & e \\ e & e \end{bmatrix} \end{aligned}$$

Many smart matrix libraries do this automatically.
This is the source of many bugs.

Broadcasting Example

Given: a $n \times 2$ matrix \mathbf{P} and a 2D column vector \mathbf{v} ,
Want: $n \times 2$ difference matrix

$$\mathbf{P} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\mathbf{P} - \mathbf{v}^T = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} - \begin{bmatrix} a & b \\ \vdots & \vdots \\ a & b \end{bmatrix} = \begin{bmatrix} x_1 - a & y_1 - b \\ \vdots & \vdots \\ x_n - a & y_n - b \end{bmatrix}$$