Arrays: computing with many numbers

Some perspective

- We have so far (mostly) looked at what we can do with single numbers (and functions that return single numbers).
- Things can get much more interesting once we allow not just one, but many numbers together.
- It is natural to view an array of numbers as one object with its own rules.
- The simplest such set of rules is that of a **vector.**

Demo "Sound as Vectors"

Vectors

A vector is an element of a Vector Space

n-vector: $$\boldsymbol{x} = \left\{ \begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right\} = [x_1 \quad x_2 \cdots x_n]^T$$

Vector space $\mathcal{V}$:

A vector space is a set $\mathcal{V}$ of vectors and a field $\mathcal{F}$ of scalars with two operations:

1) addition: $u + v \in \mathcal{V}$, and $u, v \in \mathcal{V}$

2) multiplication : $\alpha \cdot u \in \mathcal{V}$, and $u \in \mathcal{V}$, $\alpha \in \mathcal{F}$

Vector Space

The addition and multiplication operations must satisfy:

(for $\alpha, \beta \in \mathcal{F}$ and $u, v \in \mathcal{V}$)

Associativity: $u + (v + w) = (u + v) + w$

Commutativity: $u + v = v + u$

Additive identity: $v + 0 = v$

Additive inverse: $v + (-v) = 0$

Associativity wrt scalar multiplication: $\alpha \cdot (\beta \cdot v) = (\alpha \cdot \beta) \cdot v$

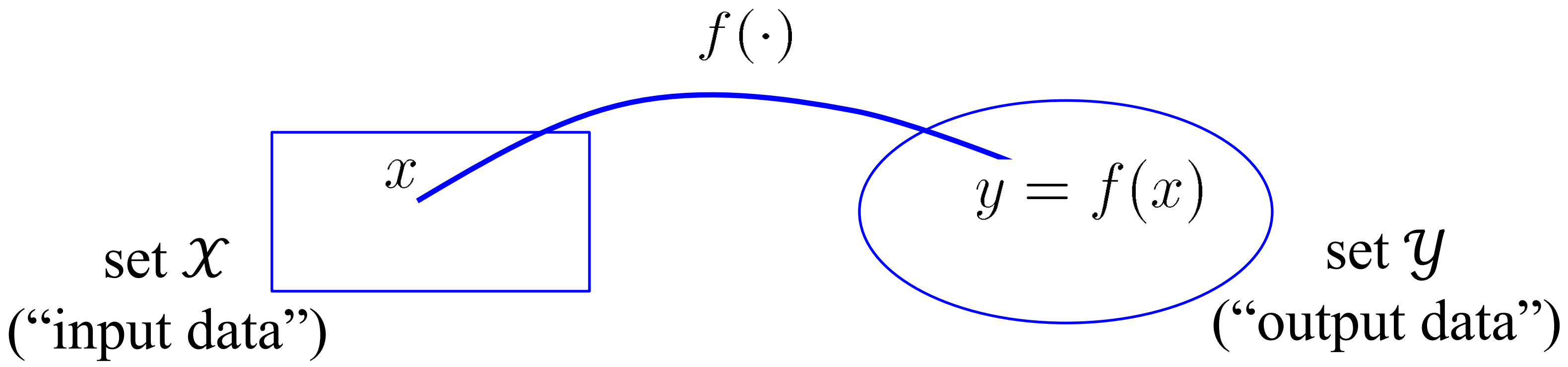
Distributive wrt scalar addition: $(\alpha + \beta) \cdot v = \alpha \cdot v + \beta \cdot v$

Distributive wrt vector addition: $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$

Scalar multiplication identity: $1 \cdot (u) = u$

Linear Functions

Function: $f : \mathcal{X} \rightarrow \mathcal{Y}$



The function f takes vectors $\boldsymbol{x} \in \mathcal{X}$ and transforms into vectors $\boldsymbol{y} \in \mathcal{Y}$

A function f is a linear function if

(1) $f(\boldsymbol{u}+\boldsymbol{v}) = f(\boldsymbol{u}) + f(\boldsymbol{v})$

(2) $f(a\boldsymbol{u}) = a\, f(\boldsymbol{u})$ for any scalar a

Iclicker question

1) Is

$$f(x) = \frac{|x|}{x}, \; f: \mathcal{R} \rightarrow \mathcal{R}$$

a linear function?

A) YES

B) NO

2) Is

$$f(x) = a\,x + b, \; f: \mathcal{R} \rightarrow \mathcal{R}, \;\; a, b \in \mathcal{R} \text{ and } a, b \neq 0$$

a linear function?

A) YES

B) NO

Matrices

- $n \times m$-matrix
$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{bmatrix}$$

- Linear functions $f(\boldsymbol{x})$ can be represented by a Matrix-Vector multiplication.

- Think of a matrix $\boldsymbol{A}$ as a linear function that takes vectors $\boldsymbol{x}$ and transforms them into vectors $\boldsymbol{y}$

$$\boldsymbol{y} = f(\boldsymbol{x}) \rightarrow \boldsymbol{y} = \boldsymbol{A}\,\boldsymbol{x}$$

- Hence we have:

$$\boldsymbol{A}\,(\boldsymbol{u} + \boldsymbol{v}) = \boldsymbol{A}\,\boldsymbol{u} + \boldsymbol{A}\,\boldsymbol{v}$$
$$\boldsymbol{A}\,(\alpha\,\boldsymbol{u}) = \alpha\,\boldsymbol{A}\,\boldsymbol{u}$$

Matrix-Vector multiplication

- Recall summation notation for matrix-vector multiplication $\boldsymbol{y} = \boldsymbol{A}\,\boldsymbol{x}$
- You can think about matrix-vector multiplication as:

Linear combination of column vectors of **A**

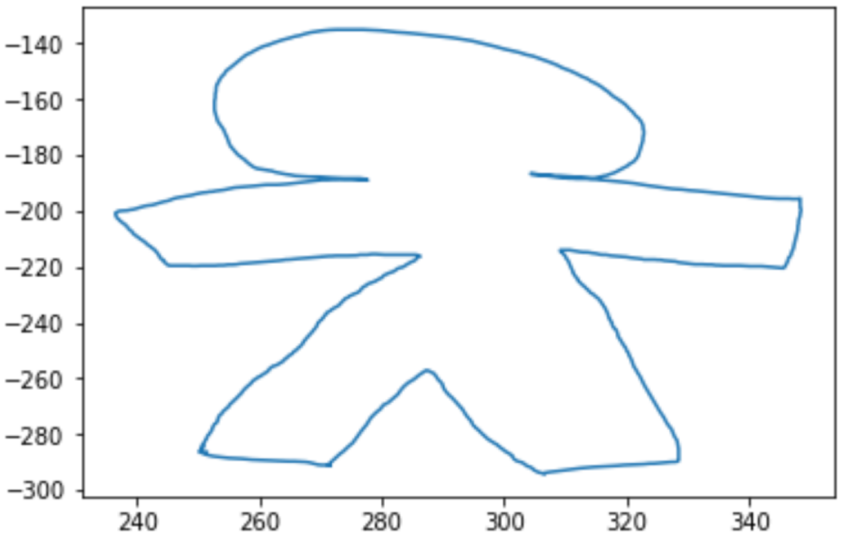
$$\boldsymbol{y} = x_1\,\mathbf{A}[:,1] + x_2\,\mathbf{A}[:,2] + \cdots + x_m\,\mathbf{A}[:,m]$$

Dot product of $\boldsymbol{x}$ with rows of **A**

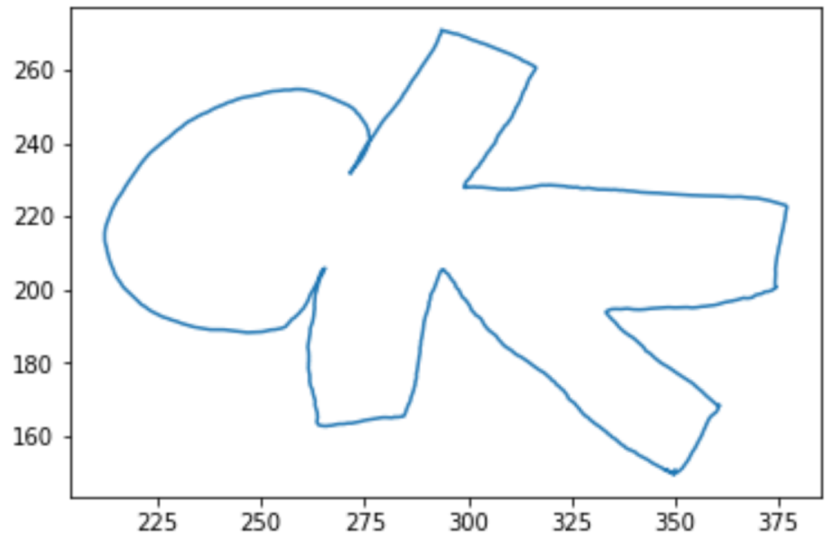
$$\boldsymbol{y} = \begin{pmatrix} \mathbf{A}[1,:] \cdot \boldsymbol{x} \\ \vdots \\ \mathbf{A}[n,:] \cdot \boldsymbol{x} \end{pmatrix}$$

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{bmatrix}$$

Matrices operating on data



Data set: $\boldsymbol{x}$



Data set: $\boldsymbol{y}$

Rotation

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$$

or

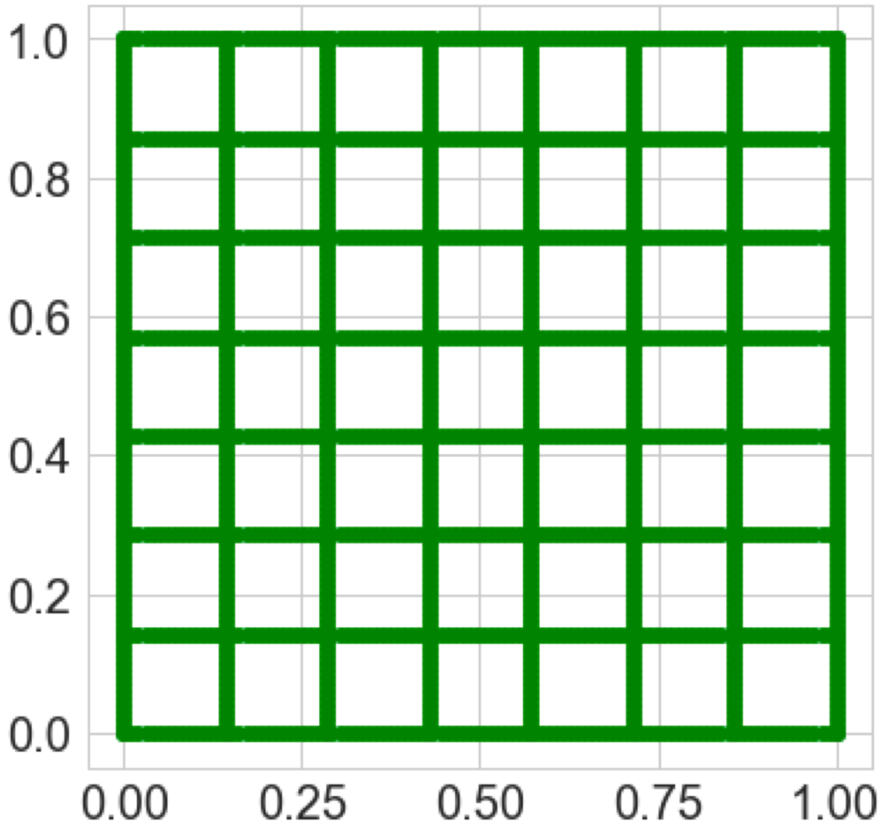
$$\boldsymbol{y} = \boldsymbol{A}\,\boldsymbol{x}$$

Example: Shear operator

Matrix-vector multiplication for each vector (point representation in 2D):

Matrices as operators

- **Data**: grid of 2D points
- Transform the data using matrix multiply



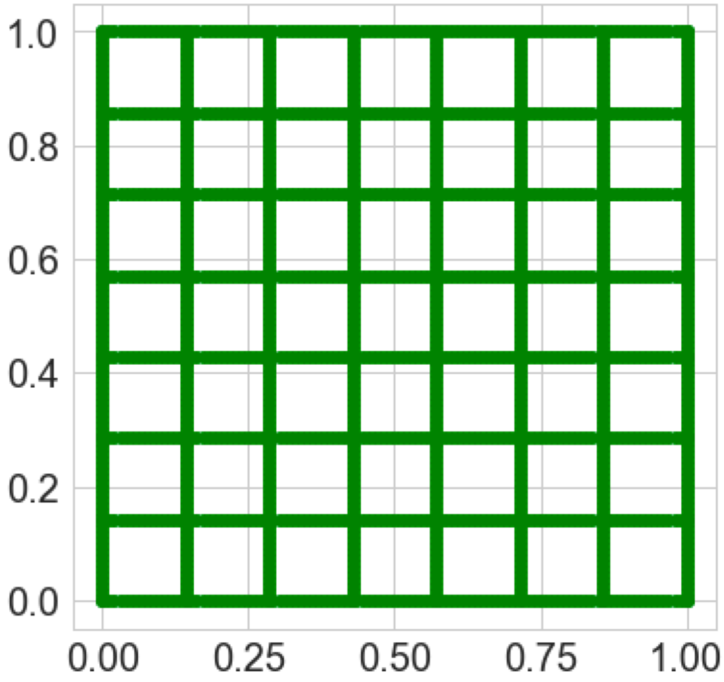
What can matrices do?

1. Shear
2. Rotate
3. Scale
4. Reflect

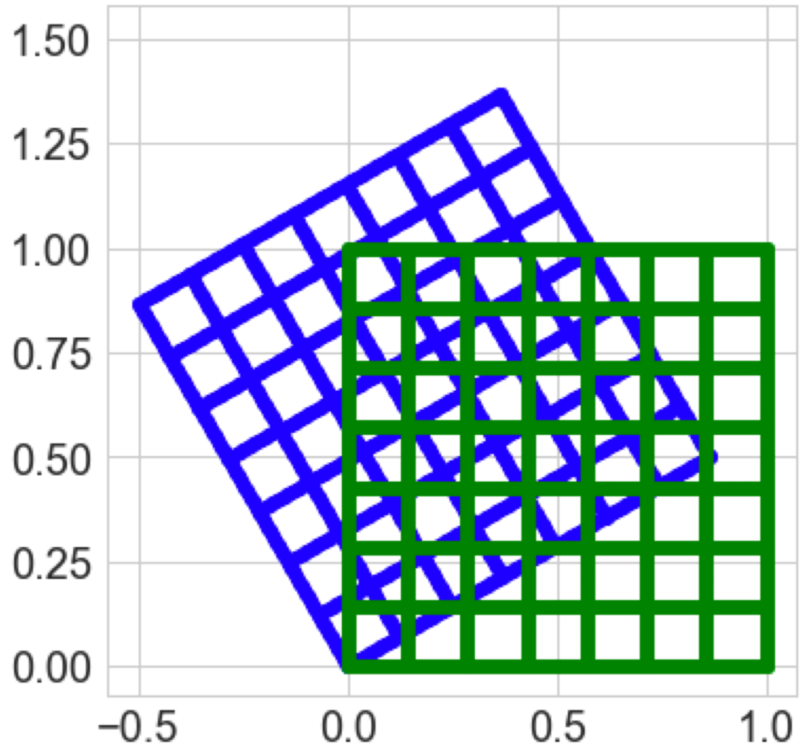
5. Can they translate?

Rotation operator

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



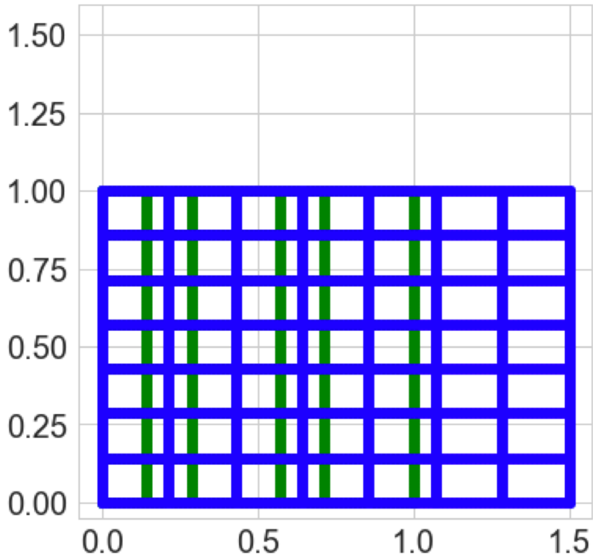
$\theta = \pi/6$



Demo “Matrix transformations”

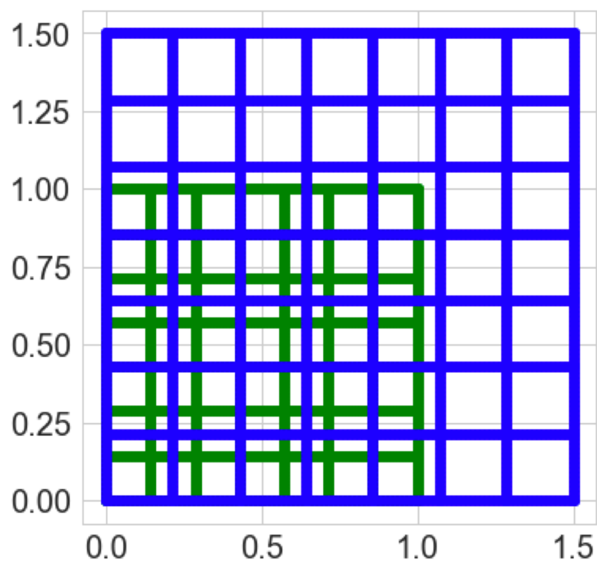
Scale operator

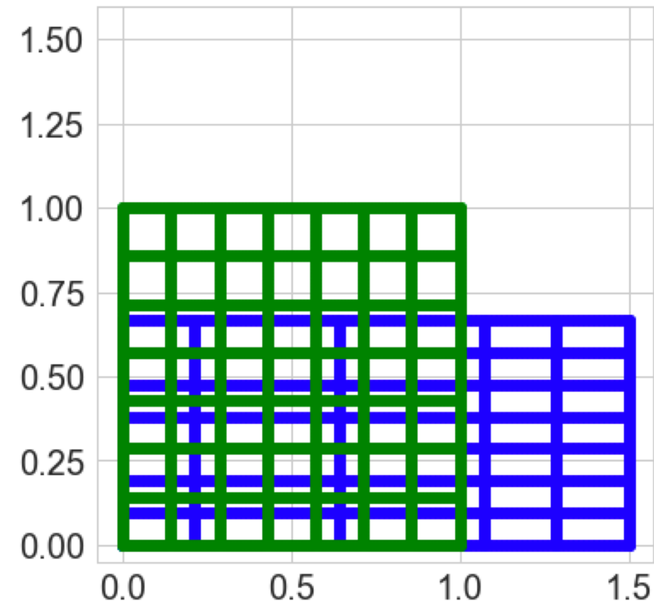
$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



$$\begin{pmatrix} 3/2 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 3/2 & 0 \\ 0 & 3/2 \end{pmatrix}$$



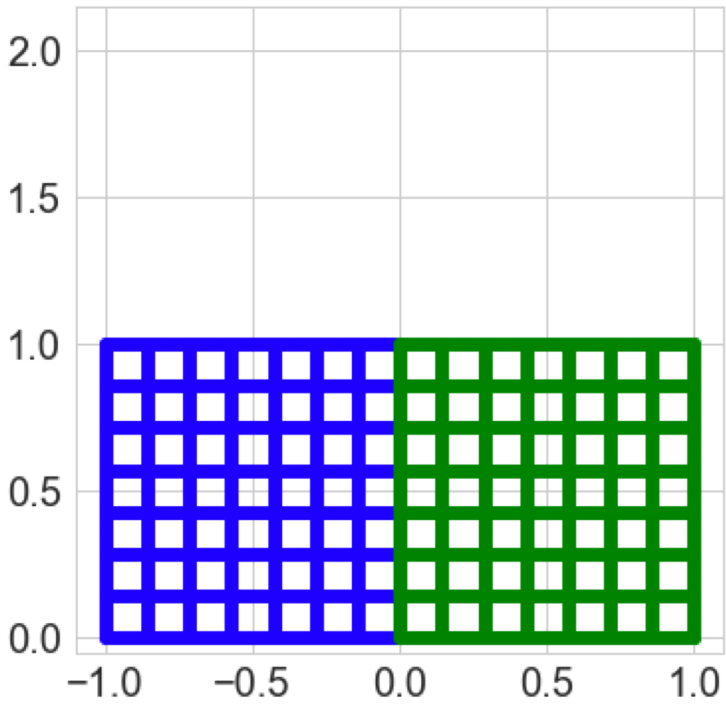


$$\begin{pmatrix} 3/2 & 0 \\ 0 & 2/3 \end{pmatrix}$$

Reflection operator

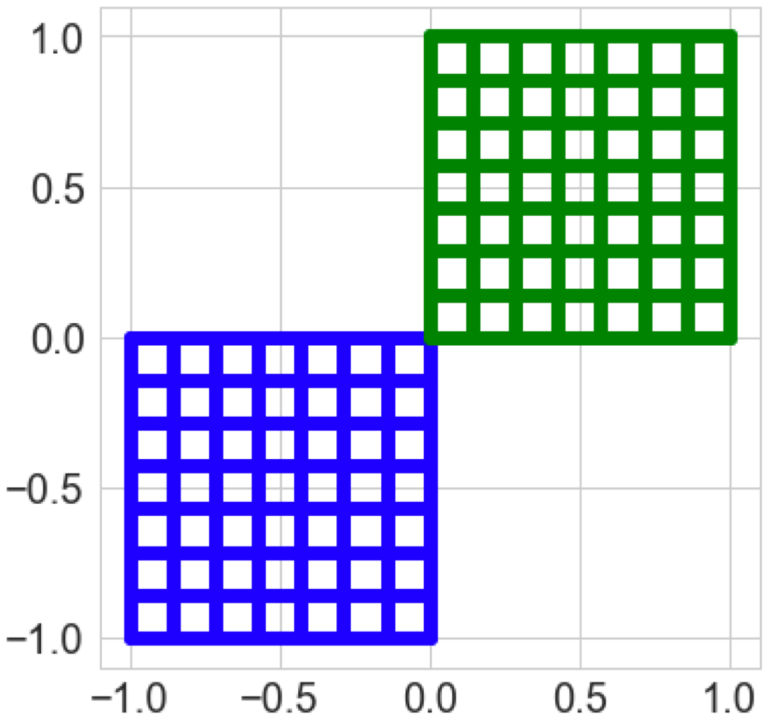
$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -a & 0 \\ 0 & -b \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$



Reflect about y-axis

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

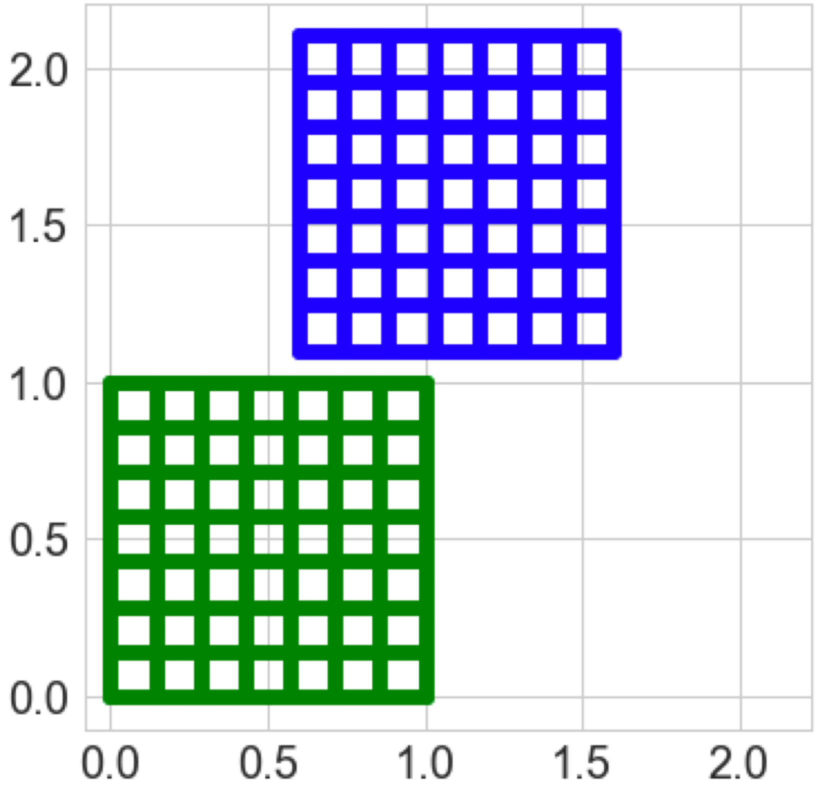


Reflect about x and y-axis

Translation (shift)

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}$$

$a = 0.6; b = 1.1$

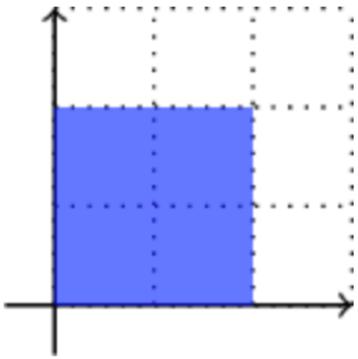


Iclicker question

Images of a brick

1 point

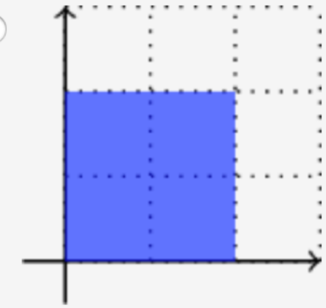
Consider the unit square in the plane:



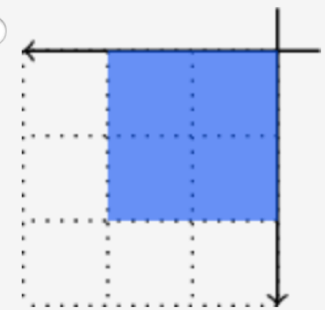
Suppose you take every vector $\mathbf{x}$ corresponding to a point in the unit square and compute $A\mathbf{x}$ for the given matrix A. Which set of points could you obtain?

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

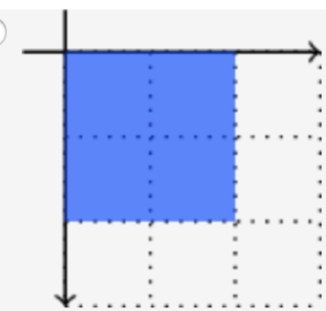
a)



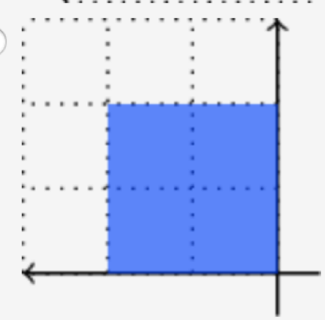
b)



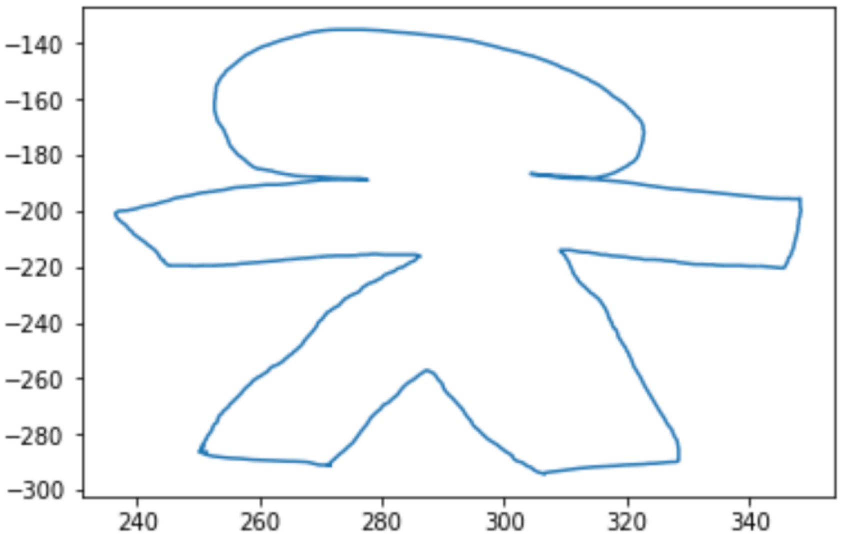
c)

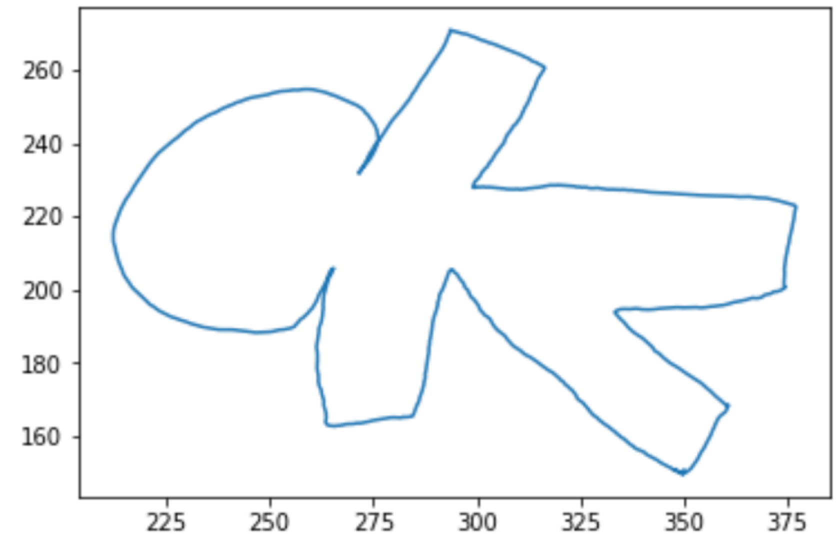


d)



Matrices operating on data





Data set: *A*

Data set: *B*

Rotation

Demo "Matrices for geometry transformation"

Notation and special matrices

- Square matrix: $m = n$

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

- Zero matrix: $A_{ij} = 0$
- Identity matrix $[\boldsymbol{I}] = [\delta_{ij}]$
- Symmetric matrix: $A_{ij} = A_{ji}$ $[\boldsymbol{A}] = [\boldsymbol{A}]^T$
- Permutation matrix:
  - Permutation of the identity matrix
  - Permutes (swaps) rows

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} c \\ a \\ b \end{pmatrix}$$

- Diagonal matrix: $A_{ij} = 0, \ \forall i, j \mid i \neq j$
- Triangular matrix:

Lower triangular: $L_{ij} = \begin{cases} L_{ij}, i \geq j \\ 0, \quad i < j \end{cases}$

Upper triangular: $U_{ij} = \begin{cases} U_{ij}, i \leq j \\ 0, \quad i > j \end{cases}$

More about matrices

- Rank: the rank of a matrix $\boldsymbol{A}$ is the dimension of the vector space generated by its columns, which is equivalent to the number of linearly independent columns of the matrix.
- Suppose $\boldsymbol{A}$ has shape $m \times n$:
  - $rank(\boldsymbol{A}) \leq \min(m, n)$
  - Matrix $\boldsymbol{A}$ is **full rank**: $rank(\boldsymbol{A}) = \min(m, n)$. Otherwise, matrix $\boldsymbol{A}$ is **rank deficient**.
- Singular matrix: a square matrix $\boldsymbol{A}$ is invertible if there exists a square matrix **B** such that $\boldsymbol{AB} = \boldsymbol{BA} = \boldsymbol{I}$. If the matrix is not invertible, it is called singular.

Iclicker question

What is the value of m that makes the matrix singular?

$$A = \begin{bmatrix} m & 2 \\ 9 & 6 \end{bmatrix}$$

A) 1

B) 3

C) 5

D) 7

Sparse Matrices

Some type of matrices contain many zeros.
Storing all those zero entries is wasteful!

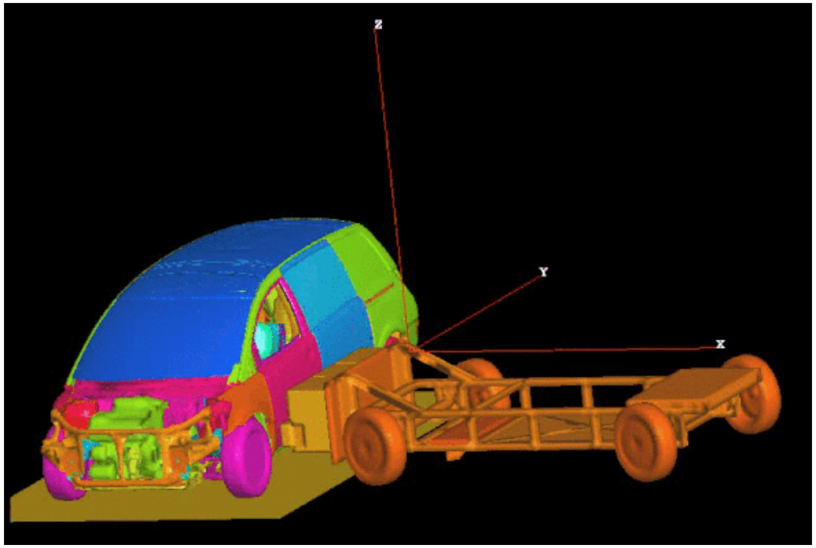
How can we efficiently store large matrices without storing tons of zeros?

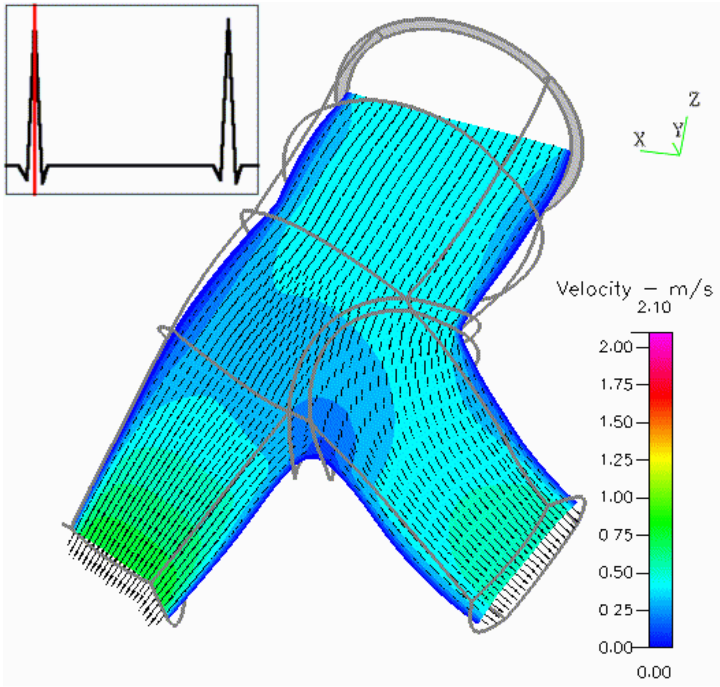
- **Sparse matrices** (vague definition): matrix with few non-zero entries.
- For practical purposes: an $m \times n$ matrix is sparse if it has $O(\min(m, n))$ non-zero entries.
- This means roughly a constant number of non-zero entries per row and column.
- Another definition: "matrices that allow special techniques to take advantage of the large number of zero elements" (J. Wilkinson)

Sparse Matrices: Goals

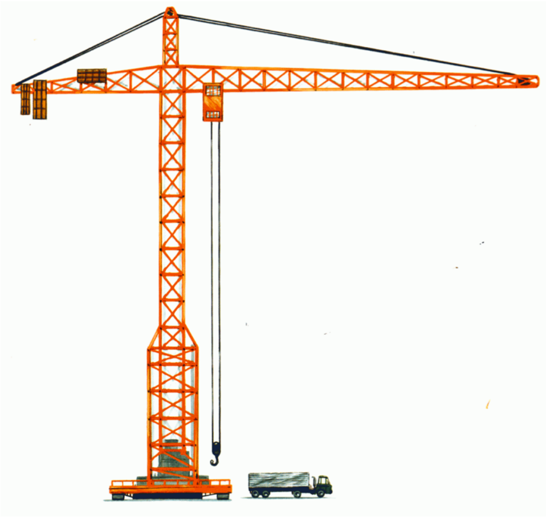
- **Perform standard matrix computations economically, i.e., without storing the zeros of the matrix.**

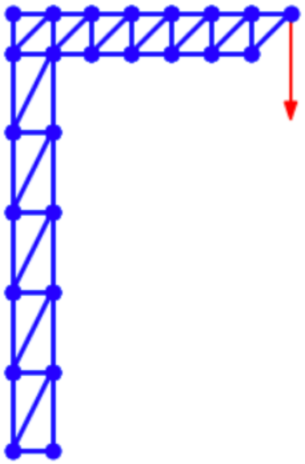
- For typical Finite Element and Finite Difference matrices, the number of non-zero entries is $O(n)$

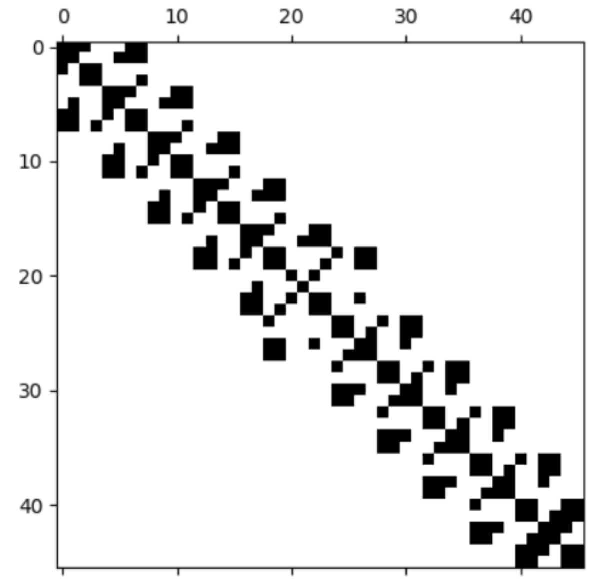




Sparse Matrices: MP example







Sparse Matrices

EXAMPLE:

Number of operations required to add two square dense matrices:

$$O(n^2)$$

Number of operations required to add two sparse matrices $\mathbf{A}$ and $\mathbf{B}$:

$$O(\mathrm{nnz}(\mathbf{A}) + \mathrm{nnz}(\mathbf{B}))$$

where $\mathrm{nnz}(\mathbf{X}) =$ number of non-zero elements of a matrix $\mathbf{X}$

Popular Storage Structures

DNS	Dense	**ELL**	Ellpack-Itpack
BND	Linpack Banded	**DIA**	Diagonal
COO	Coordinate	**BSR**	Block Sparse Row
CSR	Compressed Sparse Row	**SSK**	Symmetric Skyline
CSC	Compressed Sparse Column	**BSR**	Nonsymmetric Skyline
MSR	Modified CSR	**JAD**	Jagged Diagonal
LIL	Linked List		

note: CSR = CRS, CCS = CSC, SSK = SKS in some references

We will focus on COO and CSR!

Dense (DNS)

$$A = \begin{bmatrix} 0. & 1.9 & 0. & -5.2 \\ 0.3 & 0. & 9.1 & 0. \\ 4.4 & 5.8 & 3.6 & 0. \\ 0. & 0. & 7.2 & 2.7 \end{bmatrix}$$

$$Ashape = (nrow, ncol)$$

$$A_{dense} = [\underbrace{0. \quad 1.9 \quad 0. \quad -5.2}_{\text{Row 0}} \quad \underbrace{0.3 \quad 0. \quad 9.1 \quad 0.}_{\text{Row 1}} \quad \underbrace{4.4 \quad 5.8 \quad 3.6 \quad 0.}_{\text{Row 2}} \quad \underbrace{0. \quad 0. \quad 7.2 \quad 2.7}_{\text{Row 3}}]$$

- Simple
- Row-wise
- Easy blocked formats
- Stores all the zeros

Coordinate (COO)

$$A = \begin{bmatrix} 0. & 1.9 & 0. & -5.2 \\ 0.3 & 0. & 9.1 & 0. \\ 4.4 & 5.8 & 3.6 & 0. \\ 0. & 0. & 7.2 & 2.7 \end{bmatrix}$$

$$data = \begin{bmatrix} 1.9 & -5.2 & 0.3 & 9.1 & 4.4 & 5.8 & 3.6 & 7.2 & 2.7 \end{bmatrix}$$

$$row = \begin{bmatrix} 0 & 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \end{bmatrix}$$

$$col = \begin{bmatrix} 1 & 3 & 0 & 2 & 0 & 1 & 2 & 2 & 3 \end{bmatrix}$$

- Simple
- Does not store the zero elements
- Not sorted
- *row* and *col*: array of integers
- *data*: array of doubles

Iclicker question

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 \\ 3 & 4 & 0 & 5 & 0 \\ 6 & 0 & 7 & 8 & 9 \\ 0 & 0 & 10 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{bmatrix}$$

$$\begin{array}{lcl} data & = & [\ 12.0\ \ 9.0\ \ 7.0\ \ 5.0\ \ 1.0\ \ 2.0\ \ 11.0\ \ 3.0\ \ 6.0\ \ 4.0\ \ 8.0\ \ 10.0\ ] \\ row & = & [\ 4\ \ 2\ \ 2\ \ 1\ \ 0\ \ 0\ \ 3\ \ 1\ \ 2\ \ 1\ \ 2\ \ 3\ ] \\ col & = & [\ 4\ \ 4\ \ 2\ \ 3\ \ 0\ \ 3\ \ 3\ \ 0\ \ 0\ \ 1\ \ 3\ \ 2\ ] \end{array}$$

How many integers are stored in COO format
($\boldsymbol{A}$ has dimensions $n \times n$)?

A) nnz
B) n
C) $2\ nnz$
D) n^2
E) $2\ n$

Iclicker question

Representing a Sparse Matrix in Coordinate (COO) Form

1 point

Consider the following matrix:

$$A = \begin{bmatrix} 0 & 0 & 1.3 \\ -1.5 & 0.2 & 0 \\ 5 & 0 & 0 \\ 0 & 0.3 & 3 \\ 0 & 0 & 0 \end{bmatrix}$$

Suppose we store one row index (a 32-bit integer), one column index (a 32-bit integer), and one data value (a 64-bit float) for each non-zero entry in A. How many bytes in total are stored? Please note that 1 byte is equal to 8 bits.

A) 56 bytes
B) 72 bytes
C) 96 bytes
D) 120 bytes
E) 144 bytes

Compressed Sparse Row (CSR)

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 \\ 3 & 4 & 0 & 5 & 0 \\ 6 & 0 & 7 & 8 & 9 \\ 0 & 0 & 10 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{bmatrix}$$

$$data = [\ 1.0 \quad 2.0 \quad 3.0 \quad 4.0 \quad 5.0 \quad 6.0 \quad 7.0 \quad 8.0 \quad 9.0 \quad 10.0 \quad 11.0 \quad 12.0\ ]$$

$$col = [\ 0 \quad 3 \quad 0 \quad 1 \quad 3 \quad 0 \quad 2 \quad 3 \quad 4 \quad 2 \quad 3 \quad 4\ ]$$

$$rowptr = [\ 0 \quad 2 \quad 5 \quad 9 \quad 11 \quad 12\ ]$$

Compressed Sparse Row (CSR)

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 \\ 3 & 4 & 0 & 5 & 0 \\ 6 & 0 & 7 & 8 & 9 \\ 0 & 0 & 10 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{bmatrix}$$

$$\begin{aligned} data &= [\ 1.0\ \ 2.0\ \ 3.0\ \ 4.0\ \ 5.0\ \ 6.0\ \ 7.0\ \ 8.0\ \ 9.0\ \ 10.0\ \ 11.0\ \ 12.0\ ] \\ col &= [\ 0\ \ 3\ \ 0\ \ 1\ \ 3\ \ 0\ \ 2\ \ 3\ \ 4\ \ 2\ \ 3\ \ 4\ ] \\ rowptr &= [\ 0\ \ 2\ \ 5\ \ 9\ \ 11\ \ 12\ ] \end{aligned}$$

- Does not store the zero elements
- Fast arithmetic operations between sparse matrices, and fast matrix-vector product
- *col*: contain the column indices (array of nnz integers)
- *data*: contain the non-zero elements (array of nnz doubles)
- *rowptr*: contain the row offset (array of $n + 1$ integers)

Example - CSR format

$$A = \begin{bmatrix} 0. & 1.9 & 0. & -5.2 \\ 0. & 0. & 0. & 0. \\ 4.4 & 5.8 & 3.6 & 0. \\ 0. & 0. & 7.2 & 2.7 \end{bmatrix}$$

Norms

What's a norm?

- A generalization of 'absolute value' to vectors.
- $f(x) : \mathbb{R}^n \to \mathbb{R}_0^+$, returns a 'magnitude' of the input vector
- In symbols: Often written $\|\mathbf{x}\|$.

Define norm.

A function $\|\mathbf{x}\| : \mathbb{R}^n \to \mathbb{R}_0^+$ is called a norm if and only if

1. $\|\mathbf{x}\| > 0 \Leftrightarrow \mathbf{x} \neq \mathbf{0}$.
2. $\|\gamma\mathbf{x}\| = |\gamma| \, \|\mathbf{x}\|$ for all scalars γ.
3. Obeys triangle inequality $\|\mathbf{x} + \mathbf{y}\| \leqslant \|\mathbf{x}\| + \|\mathbf{y}\|$

Example of Norms

What are some examples of norms?

The so-called *p-norms*:

$$\left\|\begin{pmatrix} x_1 \\ \\ x_n \end{pmatrix}\right\|_p = \sqrt[p]{|x_1|^p + \cdots + |x_n|^p} \quad (p \geqslant 1)$$

$p = 1, 2, \infty$ particularly important

Unit Ball: Set of vectors $\boldsymbol{x}$ with norm $\|\boldsymbol{x}\| = 1$

Demo "Vector Norms"

Norms and Errors

If we're computing a vector result, the error is a vector.
That's not a very useful answer to 'how big is the error'.
What can we do?

Apply a norm!

How? Attempt 1:

$$\text{Magnitude of error} \neq \|\text{true value}\| - \|\text{approximate value}\|$$ **WRONG!**

Attempt 2:

$$\text{Magnitude of error} = \|\text{true value} - \text{approximate value}\|$$

Absolute and Relative Errors

What are the absolute and relative errors in approximating the location of Siebel center $(40.114, -88.224)$ as $(40, -88)$ using the 2-norm?

Absolute error:

a) *0.2240*

b) *0.3380*

c) *0.2513*

Relative error:

a) 2.59×10^{-3}

b) 2.81×10^{-3}

Matrix Norms

What norms would we apply to matrices?

- Easy answer: '*Flatten*' matrix as vector, use vector norm. This corresponds to an entrywise matrix norm called the Frobenius norm,

$$\|A\|_F := \sqrt{\sum_{i,j} a_{ij}^2}.$$

Matrix Norms

However, interpreting matrices as linear functions, what we are really interested in is the maximum amplification of the norm of any vector multiplied by the matrix,

$$\|A\| := \max_{\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\| \, .$$

These are called induced matrix norms, as each is associated with a specific vector norm $\|\cdot\|$.

Matrix Norms

The following are equivalent:

$$\max_{\|\mathbf{x}\|\neq 0} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\|\mathbf{x}\|\neq 0} \left\| A \underbrace{\frac{\mathbf{x}}{\|\mathbf{x}\|}}_{\mathbf{y}} \right\| \overset{\|\mathbf{y}\|=1}{=} \max_{\|\mathbf{y}\|=1} \|A\mathbf{y}\| = \|A\| \, .$$

Logically, for each vector norm, we get a different matrix norm, so that, e.g. for the vector 2-norm $\|\mathbf{x}\|_2$ we get a matrix 2-norm $\|A\|_2$, and for the vector ∞-norm $\|\mathbf{x}\|_\infty$ we get a matrix ∞-norm $\|A\|_\infty$.

Induced Matrix Norms

$$\|\boldsymbol{A}\|_1 = \max_j \sum_{i=1}^{n} |A_{ij}|$$

Maximum absolute column sum of the matrix $\boldsymbol{A}$

$$\|\boldsymbol{A}\|_\infty = \max_i \sum_{j=1}^{n} |A_{ij}|$$

Maximum absolute row sum of the matrix $\boldsymbol{A}$

$$\|\boldsymbol{A}\|_2 = \max_k \sigma_k$$

σ_k are the singular value of the matrix $\boldsymbol{A}$

Properties of Matrix Norms

Matrix norms inherit the vector norm properties:

1. $\|A\| > 0 \Leftrightarrow A \neq \mathbf{0}$.
2. $\|\gamma A\| = |\gamma| \, \|A\|$ for all scalars γ.
3. Obeys triangle inequality $\|A + B\| \leqslant \|A\| + \|B\|$

But also some more properties that stem from our definition:

1. $\|A\boldsymbol{x}\| \leqslant \|A\| \, \|\boldsymbol{x}\|$
2. $\|AB\| \leqslant \|A\| \, \|B\|$ (easy consequence)

Both of these are called submultiplicativity of the matrix norm.

Demo "Matrix Norms"

Iclicker question

Determine the norm of the following matrices:

1) $\left\| \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \right\|_\infty$

2) $\left\| \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \right\|_1$

a) 3

b) 4

c) 5

d) 6

e) 7

Iclicker question

Matrix Norm Approximation

Suppose you know that for a given matrix A three vectors $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ for the vector norm $||\cdot||$,

$||\mathbf{x}|| = 2$, $||\mathbf{y}|| = 1$, $||\mathbf{z}|| = 3$,

and for corresponding induced matrix norm,

$||A\mathbf{x}|| = 20$, $||A\mathbf{y}|| = 5$, $||A\mathbf{z}|| = 90$.

What is the largest lower bound for $||A||$ that you can derive from these values?

a) 90
b) 30
c) 20
d) 10
e) 5

Induced Matrix Norm of a Diagonal Matrix

What is the 2-norm-based matrix norm of the diagonal matrix

$$A = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 13 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}?$$

Induced Matrix Norm of an Inverted Diagonal Matrix

What is the 2-norm-based matrix norm of the **inverse** of the diagonal matrix

$$A = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 13 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}?$$