# Linear Algebra and DSP

Nuno Vasconcelos

UCSD

# Vector spaces

# Vector spaces

- **Definition:** a vector space is a set $\mathcal{H}$ where
  - addition and scalar multiplication are defined and satisfy:

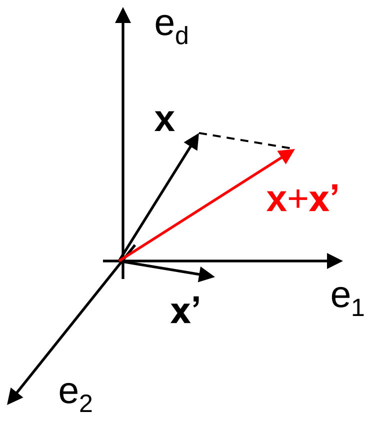
1) $x+(x'+x'') = (x+x')+x''$
2) $x+x' = x'+x \in \mathcal{H}$
3) $0 \in \mathcal{H}$, $0 + x = x$
4) $-x \in \mathcal{H}$, $-x + x = 0$

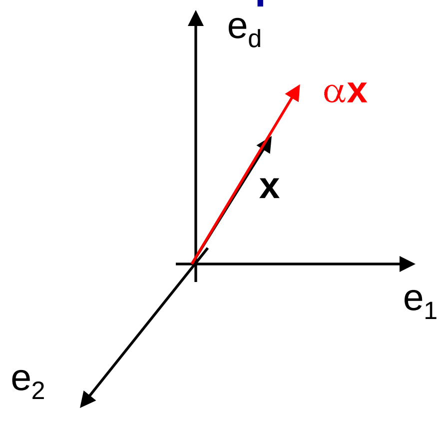
($\lambda$ = scalar; $x, x', x'' \in \mathcal{H}$)

5) $\lambda x \in \mathcal{H}$
6) $1x = x$
7) $\lambda(\lambda' x) = (\lambda\lambda')x$
8) $\lambda(x+x') = \lambda x + \lambda x'$
9) $(\lambda+\lambda')x = \lambda x + \lambda' x$

- the canonical example is $\mathbb{R}^d$ with standard

vector addition and scalar multiplication



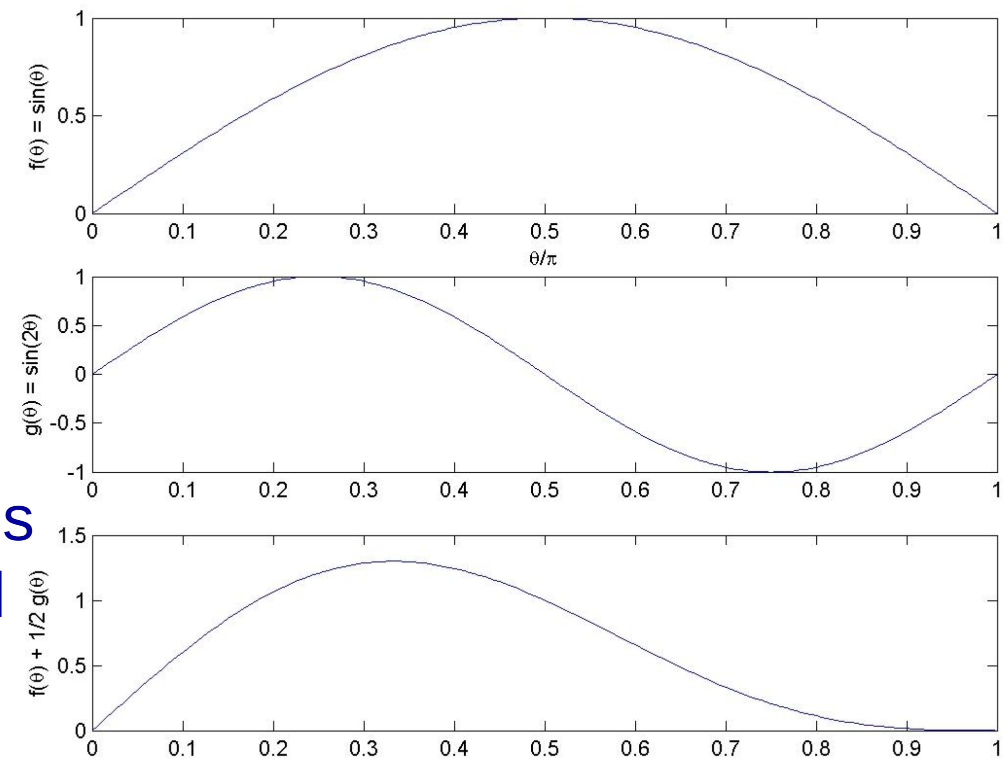



# Vector spaces

- But there are much more interesting examples
- E.g., the space of functions **f:**$\mathcal{X} \to \mathbb{R}$ with

$$(f + g)(x) = f(x) + g(x) \qquad (\lambda f)(x) = \lambda f(x)$$

- $\mathbb{R}^d$ is a vector space of finite dimension, e.g.
  - $f = ( f_1 , \dots , f_d )^T$
- When d goes to infinity we have a function
  - $f = f(t)$
- The space of all functions is an infinite dimensional vector space



# Vector spaces

- Another example is the vector space of sequences with which we work in DSP
- In 1D DSP, we represent sequences as

$$x[n], \quad n \in Z$$

- This is just a vector, which could be finite

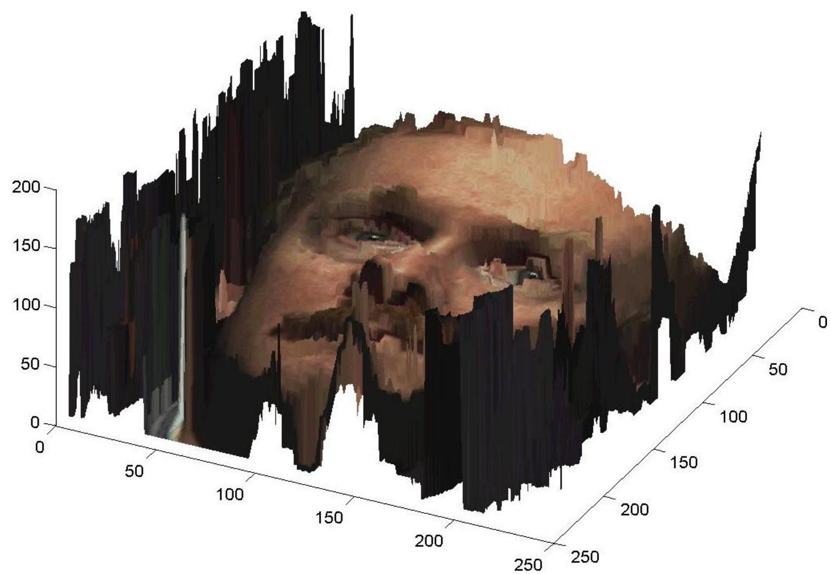
$$x = (x[1], \ldots, x[N])^T$$

or infinite

$$x = (\ldots, x[-1], x[0], x[1], \ldots)^T$$

# Data Vector Spaces

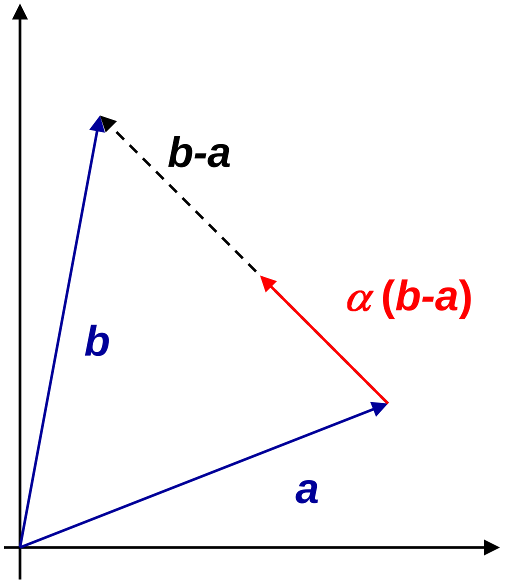
- In this course we will talk a lot about sequences
- Sequences will always be represented in a vector space:
  - A sequence is really just a point on such a space
  - from above we know how to perform basic operations on points
  - this is nice, because points can be quite abstract
  - e.g. images:
    - an image is a function on the image plane
    - it assigns a color $f(x,y)$ to each image location $(x,y)$
    - the space $\Psi$ of images is a vector space (note: assumes that images can be negative)
    - this image is a point in $\Psi$



# Images

- Because of this we can manipulate images by manipulating their vector representations
- E.g., Suppose one wants to "morph" $a(x,y)$ into $b(x,y)$:
  - One way to do this is via the path along the line from a to b.

$$c(\alpha) = a + \alpha\,(b-a)$$
$$= (1-\alpha)\,a + \alpha\,b$$



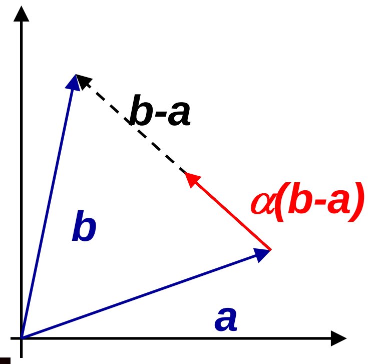

  - for $\alpha = 0$ we have $a$
  - for $\alpha = 1$ we have $b$
  - for $\alpha$ in $(0,1)$ we have a point on the line between $a$ and $b$
- To morph images we can simply apply this rule to their vector representations!

# Images

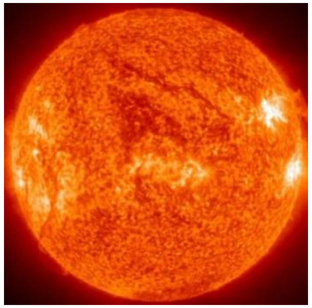
- When we make

$$c(x,y) = (1-\alpha)\, a(x,y) + \alpha\, b(x,y)$$

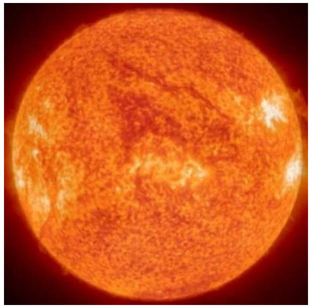
we get "image morphing":



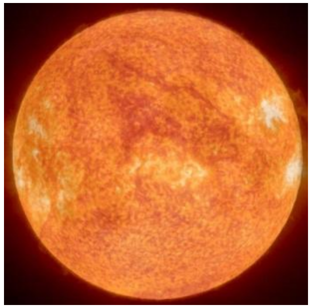

$\alpha=0$



$\alpha=0.2$



$\alpha=0.4$



$\alpha=0.6$

$\alpha=0.8$

$\alpha=1$

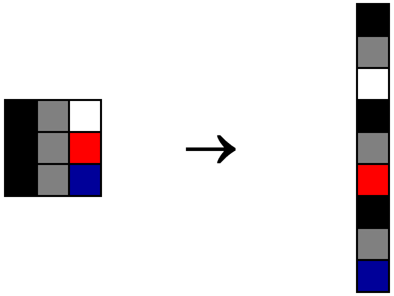
- The point is that this is possible because the images are points in a vector space.

# Images

- Images are usually represented as points in $\mathbb{R}^d$
  - Sample (discretize) an image on a finite grid to get an array of pixels

$$a(x,y) \rightarrow a(i,j)$$

  - Images are always stored like this on digital computers
  - stack all the **rows** into a vector. E.g. a *3 x 3* image is converted into a *9 x 1* vector as follows:



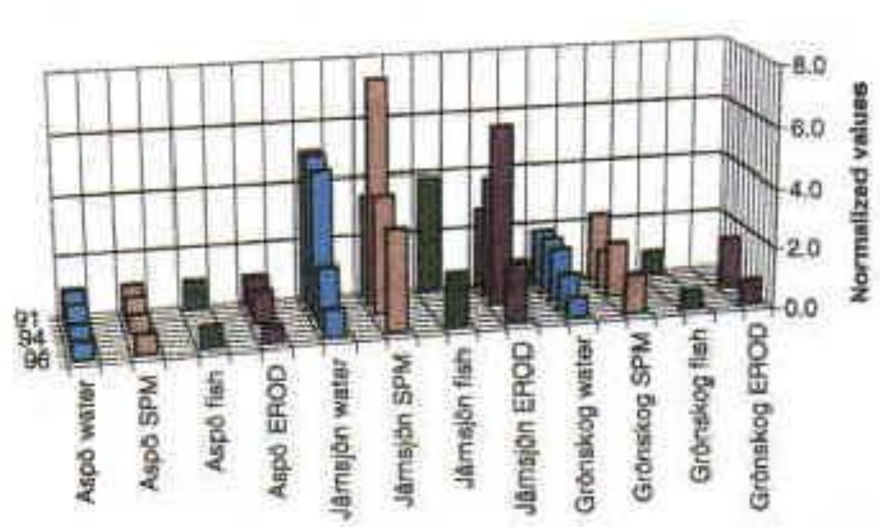
  - In general a *n x m* image vector is transformed into a *nm x 1* vector
  - Note that this is yet another vector space
- The point is that there are generally multiple different, but isomorphic, vector spaces in which the data can be represented

# Text

- Another common type of data is text
- Documents are represented by word counts:
  - associate a counter with each word
  - slide a window through the text
  - whenever the word occurs increment its counter
- This is the way search engines represent web pages

# Text

- E.g. word counts for three documents in a certain corpus (only 12 words shown for clarity




- Note that:
  - Each document is a $d$ = 12 dimensional vector
  - If I add two word count vectors (documents), I get a new word count vector (document)
  - If I multiply a word count vector (document) by a scalar, I get a word count vector
  - Note: once again we assume word counts could be negative (to make this happen we can simply subtract the average value)
- This means:
  - We are once again in a vector space (positive subset of $\mathbb{R}^d$ )
  - A document is a point in this space

# Dot-products and distances

# Bilinear forms

- Inner product vector spaces are popular because they allow us to measure similarity between data points
- The main tool for this is the inner product ("dot-product").
- We can define the dot-product using the notion of a bilinear form.
- **Definition:** a bilinear form on a real vector space $\mathcal{H}$ is a bilinear mapping

$$Q: \mathcal{H} \times \mathcal{H} \to \mathbb{R}$$
$$(x,x') \to Q(x,x')$$

"Bi-linear" means that $\forall x,x',x'' \in \mathcal{H}$

i) $Q[(\lambda x+\lambda' x'),x''] = \lambda Q(x,x'') + \lambda' Q(x',x'')$
ii) $Q[x'',(\lambda x+\lambda' x')] = \lambda Q(x'',x) + \lambda' Q(x'',x')$

# Inner Products

- **Definition:** an inner product on a real vector space $\mathcal{H}$ is a bilinear form

$$\langle \cdot,\cdot \rangle : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$$

$$(x,x') \rightarrow \langle x,x' \rangle$$

such that

i) $\langle x,x \rangle \geq 0, \ \forall x \in \mathcal{H}$

ii) $\langle x,x \rangle = 0$ if and only if $x = 0$

iii) $\langle x,y \rangle = \langle y,x \rangle$ for all $x$ and $y$

- The positive-definiteness conditions i) and ii) make the inner product a natural measure of similarity
- "nothing can be more similar to *x* than itself"
- This becomes more precise with introduction of a ***norm***

# Inner Products and Norms

- Any inner product induces a norm via

$$||x||^2 = \langle x,x \rangle$$

- By definition, any norm must obey the following properties
  - Positive-definiteness: $||x|| \geq 0$, & $||x|| = 0$ iff $x = 0$
  - Homogeneity: $||\lambda x|| = |\lambda| \, ||x||$
  - Triangle Inequality: $||x + y|| \leq ||x|| + ||y||$

- A norm defines a corresponding metric

$$d(x,y) = ||x-y||$$

which is a measure of the distance between x and y

- Always remember that the induced norm changes with a different choice of inner product!

# Inner Product

- Back to our examples:
  - In $\mathbb{R}^d$ the standard inner product is

$$\langle x, y \rangle = x^T y = \sum_{i=1}^{d} x_i y_i$$

  - Which leads to the standard Euclidean norm in $\mathbb{R}^d$

$$\|x\| = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{d} x_i^2}$$

  - The distance between two vectors is the standard Euclidean distance in $\mathbb{R}^d$

$$d(x, y) = \|x - y\| = \sqrt{(x-y)^T (x-y)} = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

# Inner Product

- In signal processing these operations have special names:
  - The inner product is the correlation between the two sequences

$$\langle x, y \rangle = x^T y = \sum_{i=1}^{d} x_i y_i$$

  - The norm is the energy of the signal

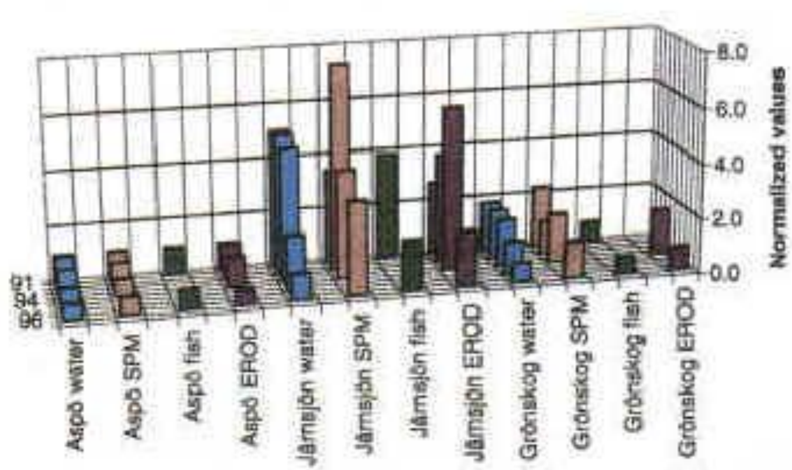
$$\|x\| = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{d} x_i^2}$$

  - The Euclidean distance is the distance

$$d(x, y) = \|x - y\| = \sqrt{(x-y)^T (x-y)} = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

# Inner Products and Norms

- Note, e.g., that this immediately gives a measure of similarity between web pages
  - compute word count vector $x_i$ from page *i*, for all *i*
  - distance between page *i* and page *j* can be simply defined as:

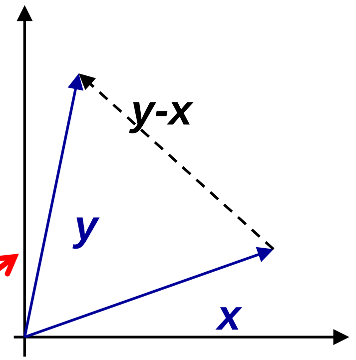


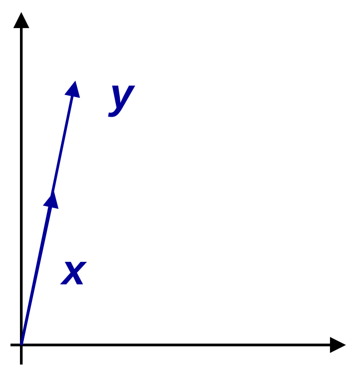

$$d(x_i, x_j) = \left\| x_i - x_j \right\| = \sqrt{(x_i - x_j)^T (x_i - x_j)}$$

  - This allows us to find, in the web, the most similar page *i* to any given page *j*.
- In fact, this is very close to the measure of similarity used by most search engines!
- What about images and other continuous valued signals?

# Inner Product

- Note that:
  - because $$\|x-y\|^2 = (x-y)^T(x-y) = x^T x - 2x^T y + y^T y = \|x\|^2 + \|y\|^2 - 2x^T y$$
  - The distance between sequences depends on
    - how much energy they have
    - their correlation.
  - The energy measures how long the vectors are
  - The correlation depends on their angle
    - *Two signals of equal energy but low correlation*
    - *Two highly correlated sequences of different energy*



# Unit vectors

- The norm measures the length of a vector
- A unit vector is a vector of norm 1
- Any vector can be made a unit vector by normalization
  - This consists of dividing the vector by its norm

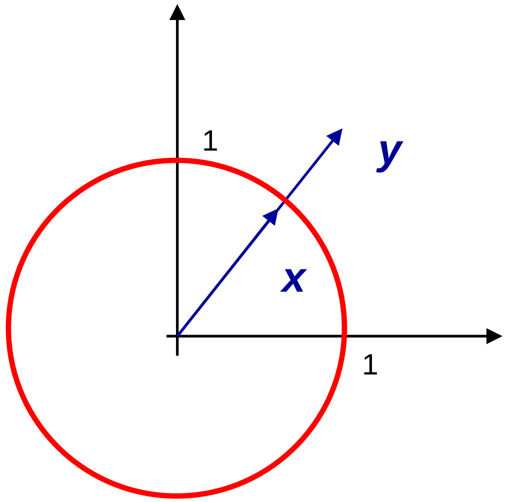
$$y \rightarrow x = \frac{y}{\|y\|}$$

  - Note that

$$\|x\| = \sqrt{x^t x} = \sqrt{\frac{y^t y}{\|y\|^2}} = 1$$

  - All unit vectors are on the unit circle

$$C = \{x \mid \|x\| = 1\}$$

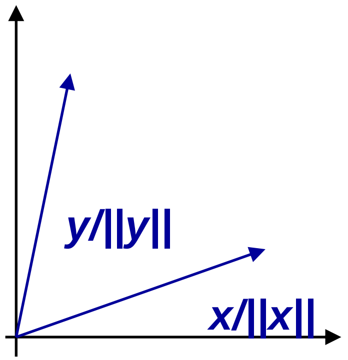


# Normalized correlation

- Is the correlation between normalized sequences

$$\rho(x,y) = \frac{x^T y}{\|x\|\|y\|} = \left(\frac{x}{\|x\|}\right)^T \frac{y}{\|y\|}$$

- And captures distance between them




$$\left\|\frac{x}{\|x\|} - \frac{y}{\|y\|}\right\|^2 = \frac{x^T x}{\|x\|^2} - 2\frac{x^T y}{\|x\|\|y\|} + \frac{y^T y}{\|y\|^2}$$
$$= 2 - 2\rho(x,y)$$

- It can be shown that

angle between x and y

$$x^T y = \|x\|\|y\|\cos(\angle x, y)$$

$$\rho(x,y) = \cos(\angle x, y)$$

# Inner Products on Function Spaces

- Recall that the space of functions is an infinite dimensional vector space
  - The standard inner product is the natural extension of that in $\mathbb{R}^d$ (just replace summations by integrals)

$$\langle f(x), g(x) \rangle = \int f(x) g(x) dx$$

  - The norm becomes the "energy" of the function

$$\| f(x) \|^2 = \int f^2(x) dx$$

  - The distance between functions the energy of the difference between them

$$d(f(x), g(x)) = \| f(x) - g(x) \|^2 = \int [f(x) - g(x)]^2 dx$$

# Inner Products on Function Spaces

- One can thus define
  - The normalized correlation between two functions

$$\rho(f(x), g(x)) = \frac{\int f(x)g(x)dx}{\int f^2(x)dx \int g^2(x)dx}$$

  - And the angle between two functions

$$\angle f(x), g(x) = \arccos\left(\frac{\int f(x)g(x)dx}{\int f^2(x)dx \int g^2(x)dx}\right)$$

# Bases

# Basis Vectors

- We know how to measure distances in a vector space
- Another interesting property is that we can fully characterize the vector space by one of its bases
- A set of vectors $x_1, \ldots, x_k$ is a basis of a vector space $\mathcal{H}$ if and only if (iff)
  - they are linearly independent

$$\sum_i c_i x_i = 0 \Leftrightarrow c_i = 0, \forall i$$

  - and they span $\mathcal{H}$: for any v in $\mathcal{H}$, v can be written as

$$v = \sum_i c_i x_i$$

- These two conditions mean that any $v \in \mathrm{H}$ can be uniquely represented in this form.

# Basis

- Note that
  - By making the vectors $x_i$ the columns of a matrix X, these two conditions can be compactly written as
  - Condition 1. The vectors $x_i$ are linear independent:

$$Xc = 0 \Leftrightarrow c = 0$$

  - Condition 2. The vectors $x_i$ span $\mathcal{H}$

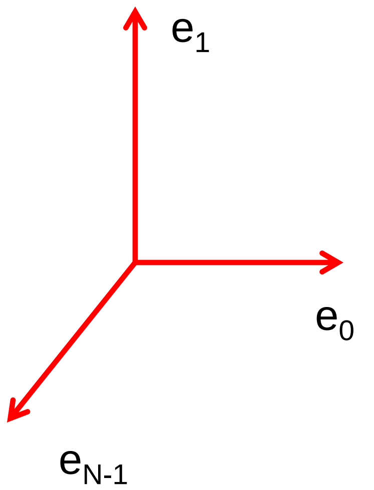
$$\forall v \neq 0, \exists c \neq 0 \text{ such that } v = Xc$$

- Also, all bases of $\mathcal{H}$ have the same number of vectors, which is called the dimension of $\mathcal{H}$
  - This is valid for any vector space!

# The canonical basis

- The simplest basis is the canonical basis

$$e_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad e_1 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \qquad \cdots \quad e_{N-1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$




- In DSP, the sequences $e_i$ are called impulse sequences

$$e_0 = \delta[n] \qquad e_1 = \delta[n-1] \qquad e_{N-1} = \delta[n-(N-1)]$$

- This is the reason why the impulse sequence has such predominance in DSP

# The impulse sequence

- For example, any sequence can be written as

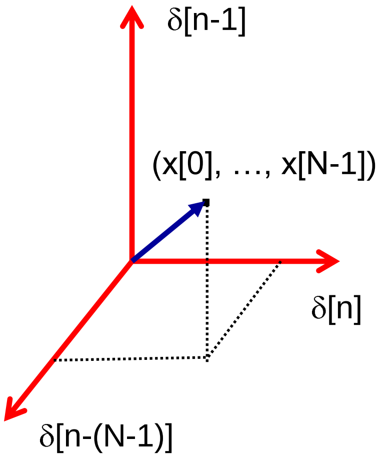
$$x[n] = \sum_k x[k]\delta[n-k]$$

- This is a well known DSP law
  - convolution with the impulse $\delta[n]$ does not change the sequence

$$x[n] = x[n] * \delta[n]$$

- Geometrically, it is just the representation of the sequence x[n] on the canonical basis

$$x[n] = x[0]\delta[n] + x[1]\delta[n-1] + \dots$$
$$+ x[N-1]\delta[n-(N-1)]$$



# The impulse sequence

- Has unit norm

$$\|\delta[n]\| = \sum_k \delta^2[k] = 1$$

- And is orthogonal to the other impulse sequences
  - For any l, m not equal

$$\langle \delta[n-l], \delta[n-m] \rangle = \sum_k \delta[k-l]\delta[k-m] = 1$$

  - Hence

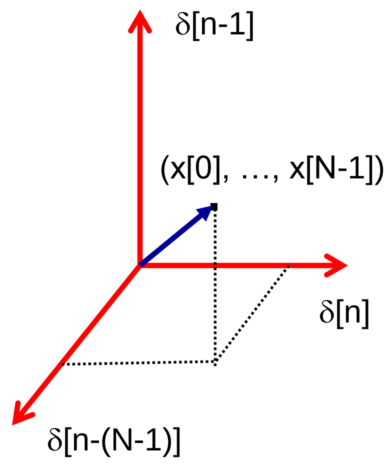
$$\angle \delta[n-l], \delta[n-m] = \frac{\pi}{2}$$

  - E.g.

Orthogonal sequences

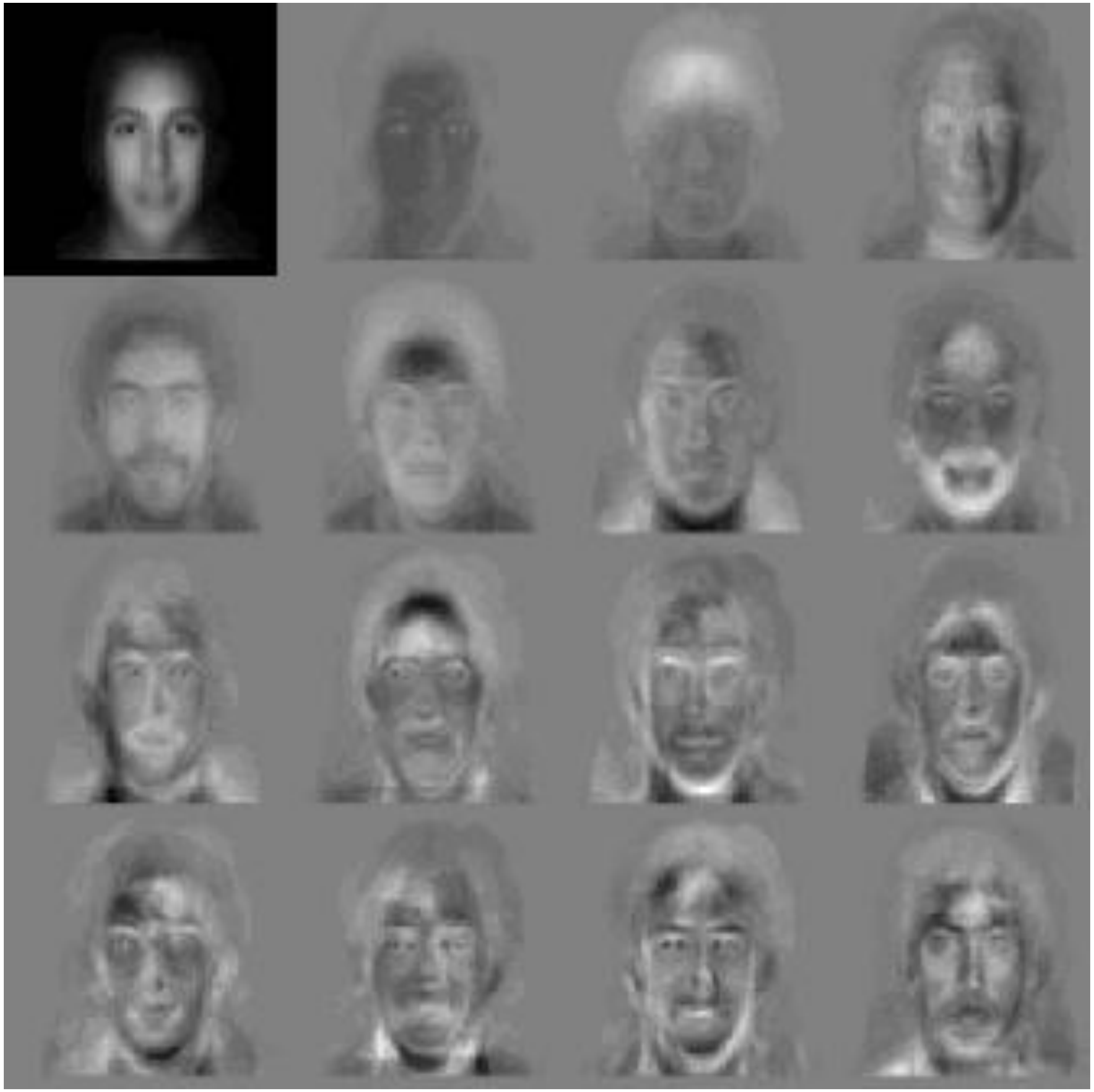
$$\delta[n-1] = (0,1,0,\ldots,0)^T$$

$$\delta[n-2] = (0,0,1,\ldots,0)^T$$



# Basis

- example
  - A basis of the vector space of images of faces
  - The figure only shows the first 16 basis vectors but there actually more
  - These vectors are orthonormal



# Orthogonality

- Two vectors are orthogonal iff their inner product is zero
  - e.g.

$$\int_0^{2\pi} \sin(ax)\cos(ax)dx = \left.\frac{\sin^2 ax}{2a}\right|_0^{2\pi} = 0$$

  in the space of functions defined on [0,2π], cos(*ax*) and sin(*ax*) are orthogonal

- Two subspaces V and W are orthogonal, $V \perp W$, if every vector in V is orthogonal to every vector in W
- a set of vectors $x_1, \ldots, x_k$ is called
  - orthogonal if all pairs of vectors are orthogonal.
  - orthonormal if all vectors also have unit norm.

$$\langle x_i, x_j \rangle = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases}$$

# Orthogonal basis

- An N-D sequence *x[n]* can be easily represented in an N-D orthogonal basis $\{b_0[n], \ldots, b_{N-1}(n)\}$
  - Sequence is a linear combination of unit basis vectors

$$x[n] = \alpha[0]\frac{b_o[n]}{\|b_o[n]\|} + \ldots + \alpha[N-1]\frac{b_{N-1}[n]}{\|b_{N-1}[n]\|}$$

  - Coefficients are the dot-products with unit basis vectors

$$\alpha[0] = \left\langle x[n], \frac{b_o[n]}{\|b_o[n]\|} \right\rangle \quad \cdots \quad \alpha[N-1] = \left\langle x[n], \frac{b_{N-1}[n]}{\|b_{N-1}[n]\|} \right\rangle$$

# Orthogonal basis

- Note that for the impulse basis, $b_m[n] = \delta[n-m]$
  - The basis vectors are already unit vectors
  - The sequence is

$$x[n] = \alpha[0]b_o[n] + \ldots + \alpha[N-1]b_{N-1}[n]$$
$$= \alpha[0]\delta[n] + \ldots + \alpha[N-1]\delta[n-(N-1)]$$
$$= \sum_{k} \alpha[k]\delta[n-k]$$

  - The coefficients are

$$\alpha[k] = \langle x[n], b_k[n] \rangle = \langle x[n], \delta[n-k] \rangle = x[k]$$

  - And we are back to the fundamental formula

$$x[n] = \sum_{k} x[k]\delta[n-k]$$

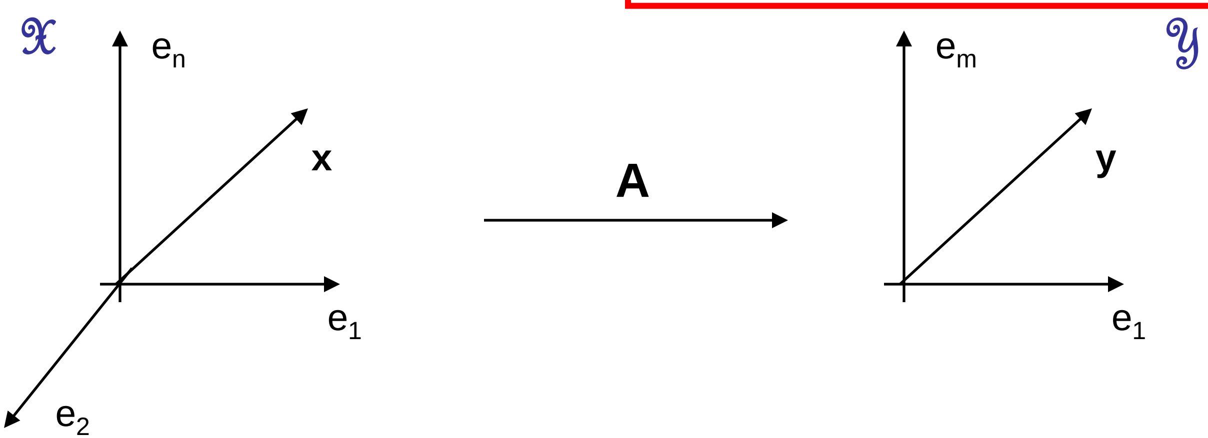
# Matrices and LTI systems

# Matrix

- an m x n matrix represents a linear operator that maps a vector from the *domain* $\mathcal{X} = \mathbf{R}^n$ to a vector in the codomain $\mathcal{Y} = \mathbf{R}^m$

- E.g. the equation $y = Ax$ sends $x$ in $\mathbf{R}^n$ to $y$ in $\mathbf{R}^m$ according to $\Longrightarrow$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$




- note that there is nothing magical about this, it follows rather mechanically from the definition of matrix-vector multiplication

# Linear systems

- Hence, a square matrix can be used to represent a linear system
- domain $\mathfrak{X} = \mathbf{R}^N$ is the vector space of input sequences
- codomain $\mathfrak{Y} = \mathbf{R}^N$ is the vector space of output sequences
- Note that if the input is a linear combination of sequences

$$x[n] = ax_1[n] + bx_2[n]$$

- The output is the same linear combination of the corresponding outputs

$$y[n] = Ax[n] = A(ax_1[n] + bx_2[n])$$

$$= aAx_1[n] + bAx_2[n] = ay_1[n] + by_2[n]$$

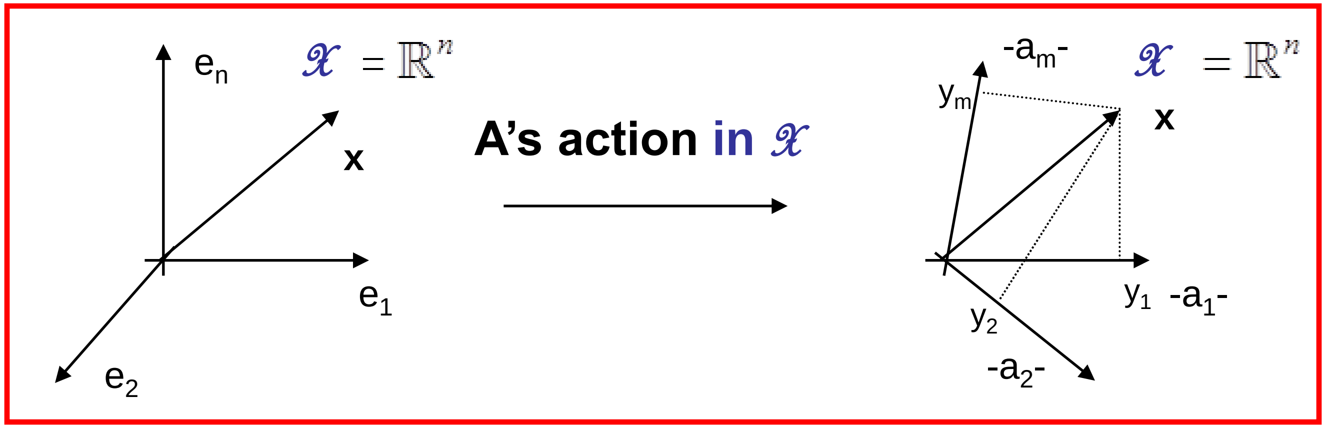
- This is the definition of linear system

# Matrix-Vector Multiplication I

- Consider $y = Ax$, i.e. $y_i = \sum_{j=1}^{n} a_{ij}x_j$ , $i = 1,\ldots,m$
- We can think of this as

$$\begin{bmatrix} \vdots \\ y_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & & \vdots \\ a_{i1} & \cdots & a_{in} \\ \vdots & & \vdots \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \vdots \\ \sum_{j=1}^{n} a_{ij} x_j \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ -a_i - \quad x \\ \vdots \end{bmatrix}$$ ($m$ rows)

- where "$(- a_i -)$" means the $i^{th}$ row of $A$. Hence
  - the $i^{th}$ component of $y$ is the inner product of $(- a_i -)$ and $x$.
  - y is the projection of x on the subspace (of the domain space) spanned by the rows of A

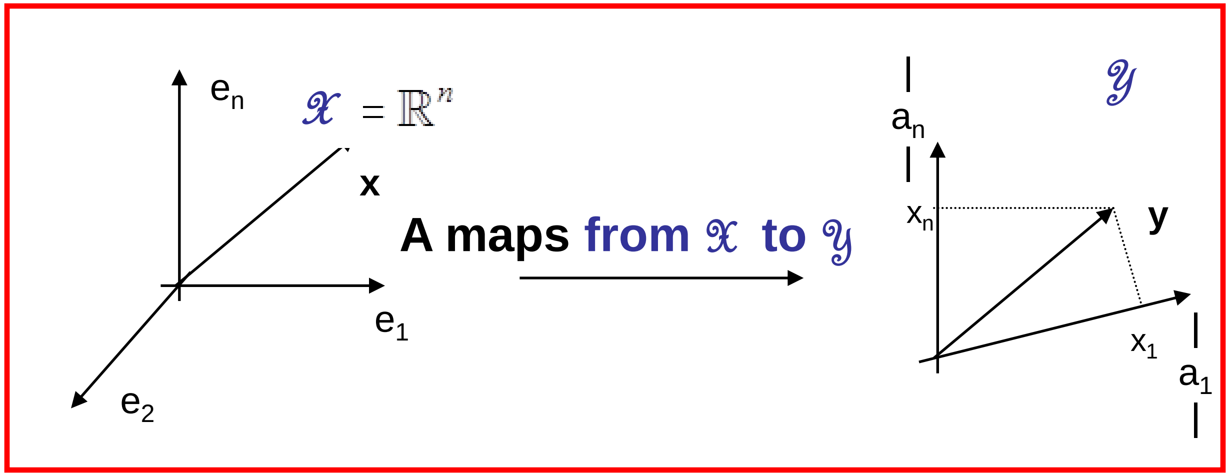


# Matrix-Vector Multiplication II

- But there is more. Let $y = Ax$, i.e. $y_i = \sum_{j=1}^{n} a_{ij}x_j$ , now be written as

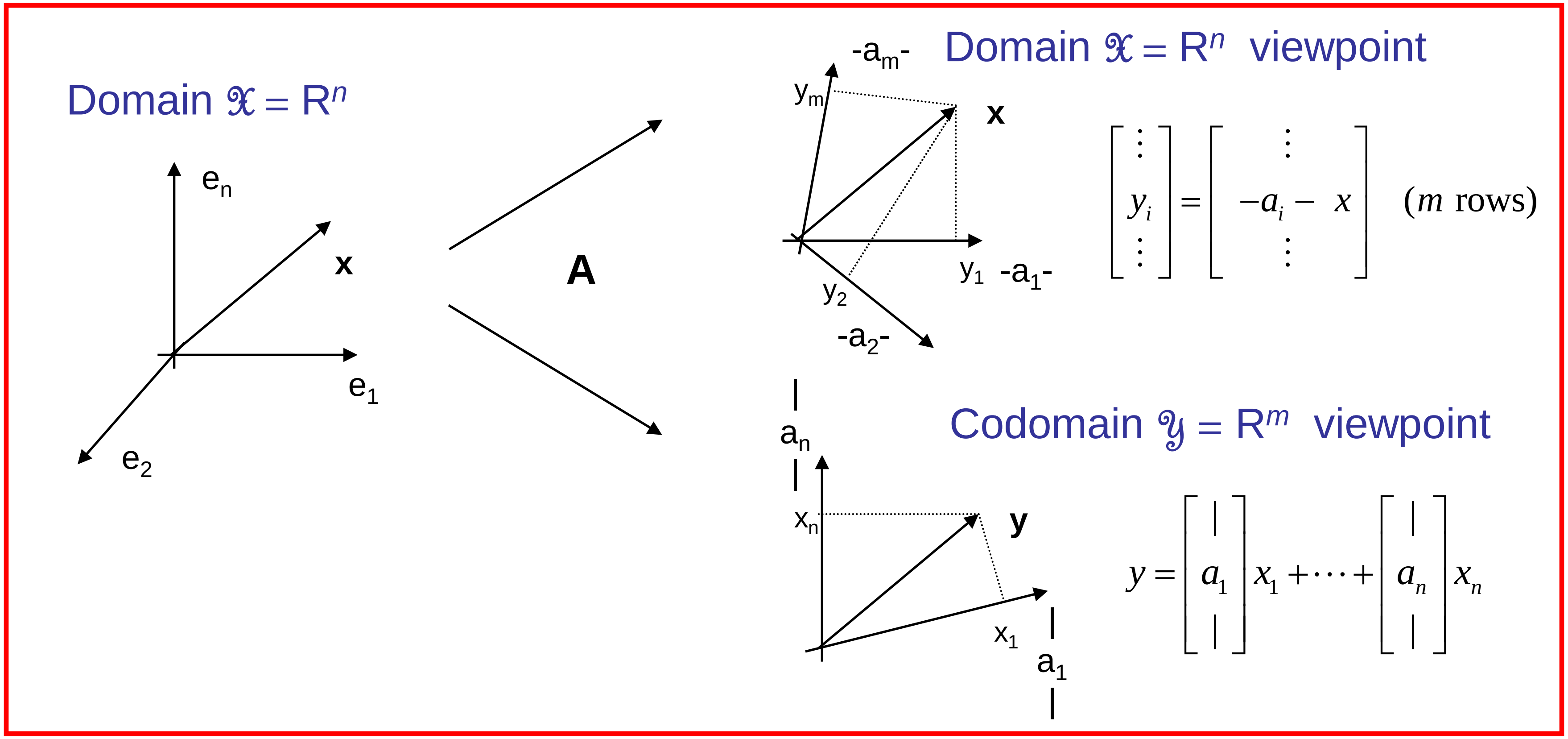
$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \vdots \\ \sum_{j=1}^{n} a_{ij}x_j \\ \vdots \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + \cdots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} x_1 + \cdots + \begin{bmatrix} | \\ a_n \\ | \end{bmatrix} x_n$$

- where $a_i$ with "|" above and below means the $i^{th}$ column of $A$.
- hence
  - $x_i$ is the $i^{th}$ component of $y$ in the subspace (of the co-domain) spanned by the columns of A
  - y is a linear combination of the columns of A



# Matrix-Vector Multiplication

- two alternative (dual) pictures of $y = Ax$:
  - y = coordinates of x in row space of A (The $\mathfrak{X} = \mathrm{R}^n$ viewpoint)




$$\begin{bmatrix} \vdots \\ y_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ -a_i- \\ \vdots \end{bmatrix} x \quad (m \text{ rows})$$

$$y = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} x_1 + \cdots + \begin{bmatrix} | \\ a_n \\ | \end{bmatrix} x_n$$

  - x = coordinates of y in column space of A ($\mathfrak{Y} = \mathrm{R}^m$ viewpoint)

# A cool trick

- the matrix multiplication formula

$$C = AB \Leftrightarrow c_{ij} = \sum_k a_{ik} b_{kj}$$

also applies to "block matrices" when these are defined properly

- for example, if *A,B,C,D,E,F,G,H* are matrices,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

- only but important caveat: the sizes of A,B,C,D,E,F,G,H have to be such that the intermediate operations make sense! (they have to be "conformal")

# Matrix-Vector Multiplication

- This makes it easy to derive the two alternative pictures
- The row space picture (or viewpoint):

$$\begin{bmatrix} \vdots \\ y_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & & \vdots \\ a_{in} & \cdots & a_{in} \\ \vdots & & \vdots \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \vdots \\ (-a_i-)_{1xn} \\ \vdots \end{bmatrix} x_{nx1} = \begin{bmatrix} \vdots \\ (-a_i-)x \\ \vdots \end{bmatrix}$$

is just like scalar multiplication, with *blocks* $(-a_i-)$ and $x$

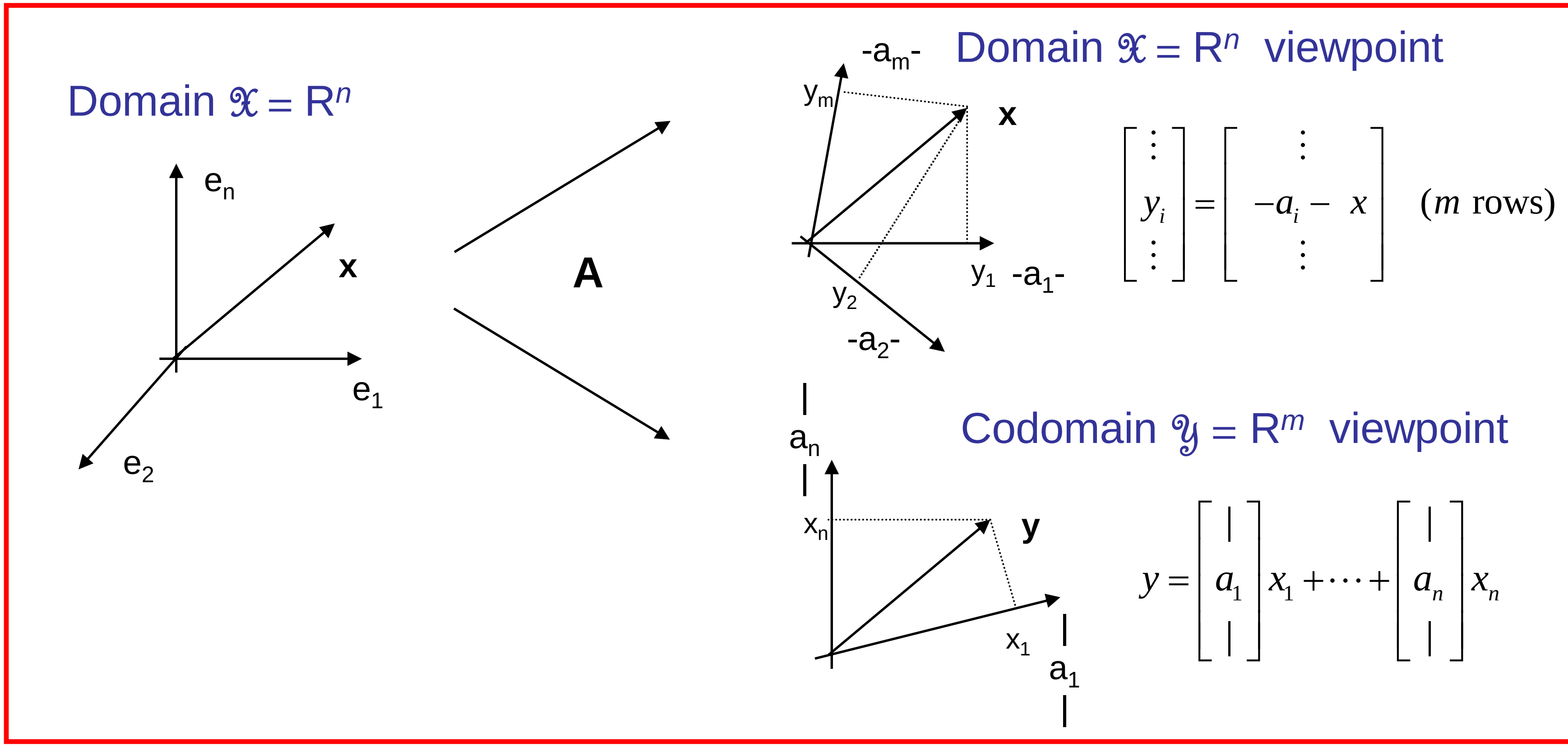
- The column space picture (or viewpoint):

$$\begin{bmatrix} \vdots \\ y_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & & \vdots \\ a_{in} & \cdots & a_{in} \\ \vdots & & \vdots \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \\ {}_{mx1} & & {}_{mx1} \end{bmatrix} \begin{bmatrix} (x_1)_{1x1} \\ \vdots \\ (x_n)_{1x1} \end{bmatrix} = \sum_i \begin{pmatrix} | \\ a_i \\ | \end{pmatrix} x_i$$

is just a inner product, with (scalar) blocks $x_i$ and the column blocks of $A$.

# Matrix-Vector Multiplication

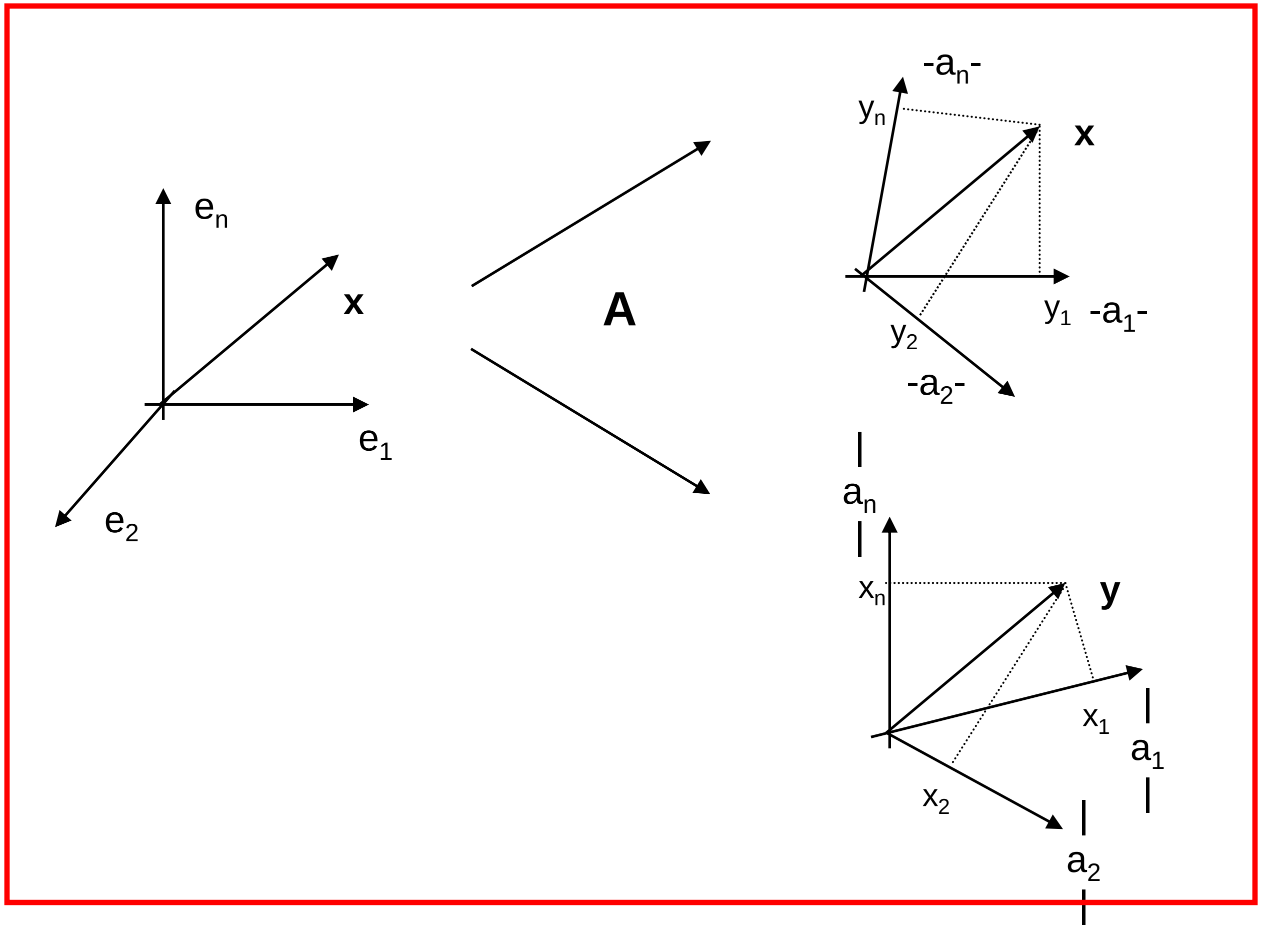
- two alternative (dual) pictures of $y = Ax$:
  - y = coordinates of x in row space of A (The $\mathfrak{X} = R^n$ viewpoint)




  - x = coordinates of y in column space of A ($\mathfrak{Y} = R^m$ viewpoint)

# Square *n* x *n* matrices

- in this case $m = n$ and the row and column subspaces are *both* equal to (copies of) $R^n$



# LTI systems

- A is time invariant when
  - x[n] has response y[n]
  - If and only if, for any m, x[n-m] has response y[n-m]

$$y[n] = Ax[n] \Leftrightarrow y[n-m] = Ax[n-m]$$

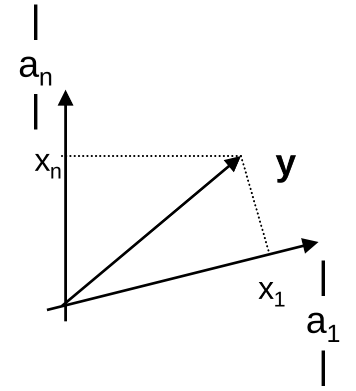
  - How does this constrain A?
    - Let $x[n] = \delta[n]$
    - call the output impulse response

$$h[n] = A\delta[n]$$

    - Using the codomain viewpoint

$$h[n] = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} \times 1 + \begin{bmatrix} | \\ a_2 \\ | \end{bmatrix} \times 0 + \ldots + \begin{bmatrix} | \\ a_{N-1} \\ | \end{bmatrix} \times 0 = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix}$$

"Impulse response is first column of A"




$$y = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} x_1 + \cdots + \begin{bmatrix} | \\ a_n \\ | \end{bmatrix} x_n$$

# LTI systems

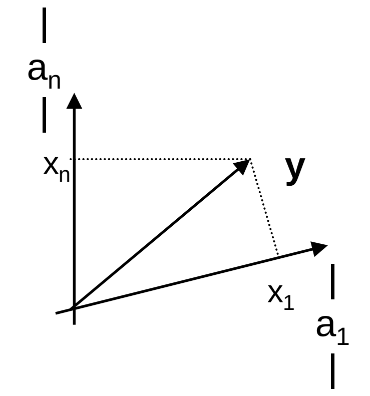
- A is time invariant when
  - Let $x[n] = \delta[n-1]$
  - the output is shifted impulse response

$$h[n-1] = A\delta[n-1]$$

  - Using the codomain viewpoint

$$h[n-1] = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} \times 0 + \begin{bmatrix} | \\ a_2 \\ | \end{bmatrix} \times 1 + \ldots + \begin{bmatrix} | \\ a_{N-1} \\ | \end{bmatrix} \times 0 = \begin{bmatrix} | \\ a_2 \\ | \end{bmatrix}$$

"h[n-1] is second column of A"




$$y = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} x_1 + \cdots + \begin{bmatrix} | \\ a_n \\ | \end{bmatrix} x_n$$

- Repeating this for all shifts of the input we have
  – The $k^{th}$ column of A is h[n-k]!

# LTI systems

- The matrix A has the structure

$$A = \begin{bmatrix} | & | & & | \\ h[n] & h[n-1] & \ldots & h[n-(N-1)] \\ | & | & & | \end{bmatrix}$$

- Columns are shifts of the impulse response
  - Check:
  - what if impulse response is the impulse?
  - A is the identity matrix
  - Ax = x for all x
  - The system does not change the input
  - It is an all-pass filter

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

# LTI systems

- What are the rows?
  - Note that we can write

$$A = \begin{bmatrix} h[0] & h[-1] & & h[-N+1] \\ h[1] & h[0] & & h[-N+2] \\ \vdots & \vdots & \ddots & \\ h[N-1] & h[N-2] & & h[0] \end{bmatrix}$$

  - the $k^{th}$ row of Ax is

$$[h[k] \quad h[k-1] \quad \ldots \quad h[k-(N-1)]]$$

  - This is the sequence

$$g_k[n] = h[k-n]$$

  - Obtained by flipping h and shifting by k

# LTI systems

- Hence, we have

Domain $\mathfrak{X}$ viewpoint

$$A = \begin{bmatrix} - & g_0[n] & - \\ & \vdots & \\ - & g_{N-1}[n] & - \end{bmatrix} = \begin{bmatrix} - & h[-n] & - \\ & \vdots & \\ - & h[N-1-n] & - \end{bmatrix}$$

$$\begin{bmatrix} \vdots \\ y_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ (-a_i-)x \\ \vdots \end{bmatrix}$$

- And two views of an LTI system

Codomain $\mathfrak{Y}$ viewpoint

$$A = \begin{bmatrix} | & | & & | \\ h[n] & h[n-1] & \ldots & h[n-(N-1)] \\ | & | & & | \end{bmatrix}$$

$$y = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} x_1 + \cdots + \begin{bmatrix} | \\ a_n \\ | \end{bmatrix} x_n$$

# Convolution

- Two ways to compute the output
- Under the domain viewpoint

$$\begin{bmatrix} y[0] \\ \vdots \\ y[N-1] \end{bmatrix} = \begin{bmatrix} - & h[-n] & - \\ & \vdots & \\ - & h[N-1-n] & - \end{bmatrix} x[n]$$

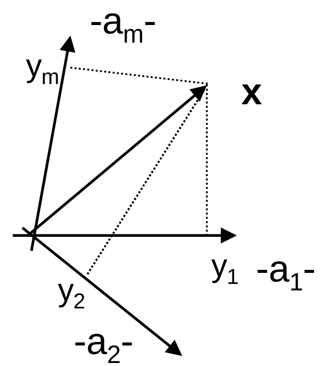
$$= \begin{bmatrix} \langle h[-n], x[n] \rangle \\ \vdots \\ \langle h[N-1-n], x[n] \rangle \end{bmatrix}$$

- We obtain the convolution formula

$$y[k] = \langle h[k-n], x[n] \rangle = \sum_n h[k-n]x[n]$$

Domain $\mathfrak{X}$ viewpoint

$$\begin{bmatrix} \vdots \\ y_i \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ - & a_i & - \\ & \vdots & \end{bmatrix} x \quad (m \text{ rows})$$



# Convolution

- Under the codomain viewpoint

$$\begin{bmatrix} | \\ y[n] \\ | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ h[n] & h[n-1] & \dots & h[n-(N-1)] \\ | & | & & | \end{bmatrix} \begin{bmatrix} | \\ x[n] \\ | \end{bmatrix}$$

$$= \begin{bmatrix} | \\ h[n] \\ | \end{bmatrix} x[0] + \dots + \begin{bmatrix} | \\ h[n-(N-1)] \\ | \end{bmatrix} x[N-1]$$

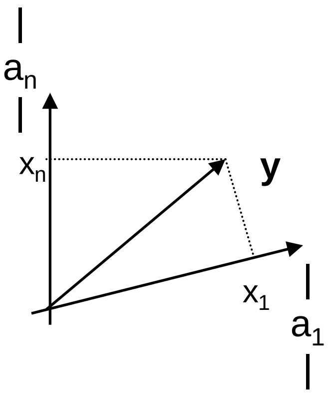
- We obtain the alternative view of convolution

$$y[n] = \sum_k h[n-k]x[k]$$

- Note that the formulas are the same, but interpretation is different

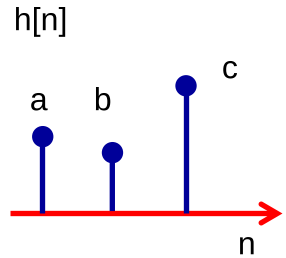
Codomain $\mathfrak{X}$ viewpoint

$$y = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} x_1 + \cdots + \begin{bmatrix} | \\ a_n \\ | \end{bmatrix} x_n$$

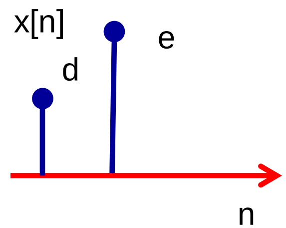


# Example

- Impulse response:

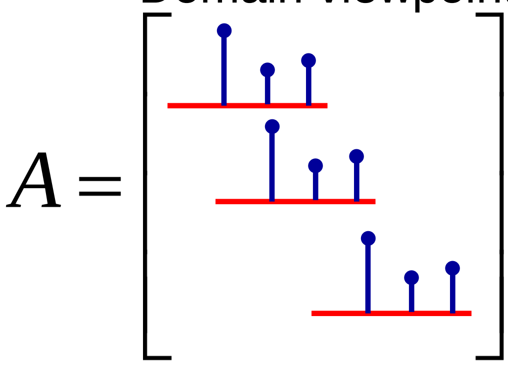



- Input:

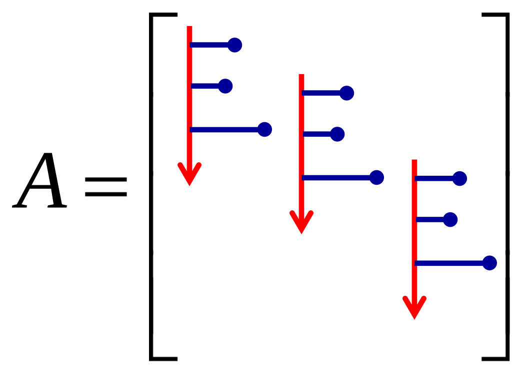



- System:

Domain viewpoint

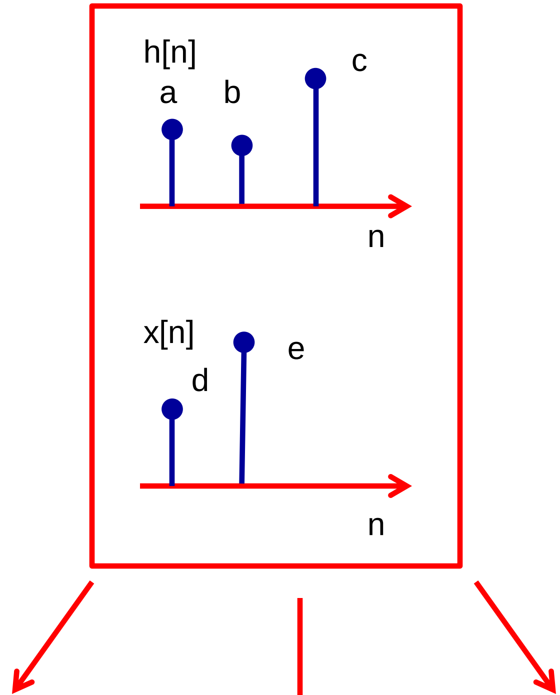



Codomain viewpoint

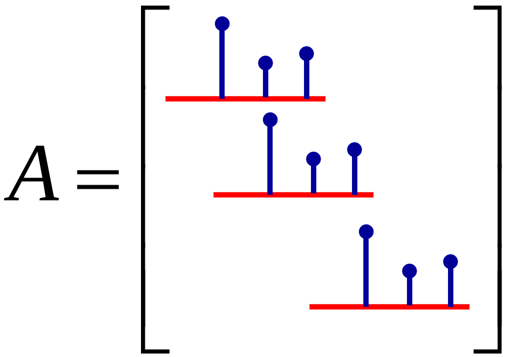


# Example

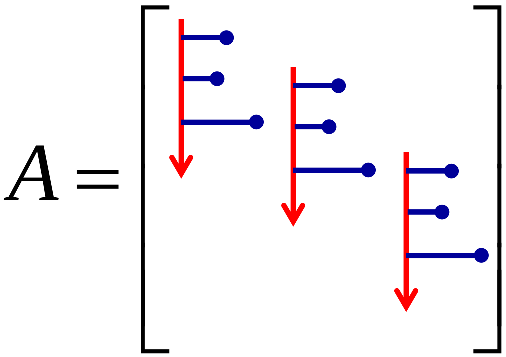
- convolution:

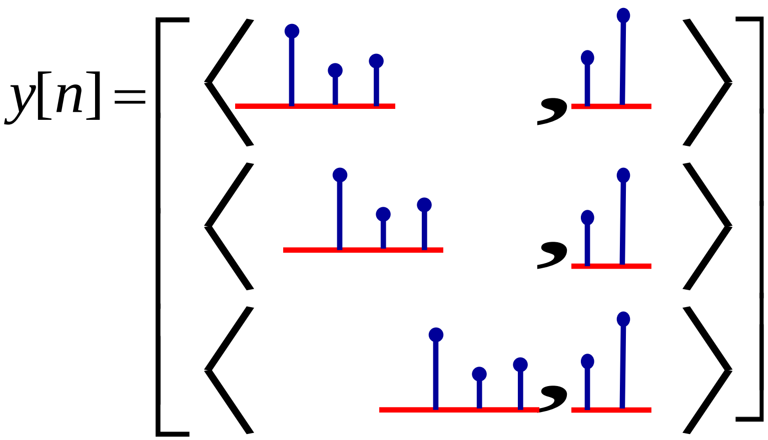


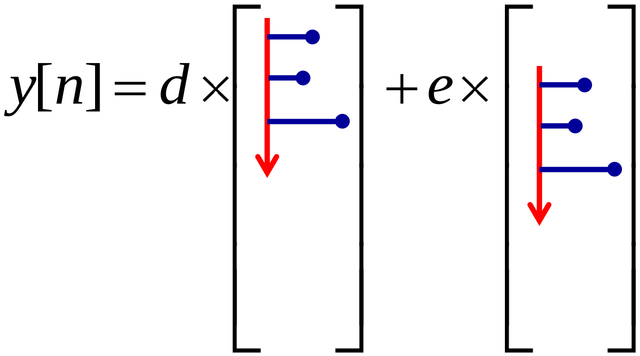

Domain viewpoint




Codomain viewpoint



# The fundamental spaces

# Orthogonal matrices

- A matrix is called orthogonal if it is square and has orthonormal columns.
- Important properties:
  - 1) The inverse of an orthogonal matrix is its transpose
    - this can be easily shown with the block matrix trick. (Also see later.)

$$A^T A = \begin{bmatrix} \vdots \\ -a_i^T- \\ \vdots \end{bmatrix}_{1\times n} \begin{bmatrix} \cdots & \begin{pmatrix} | \\ a_j \\ | \end{pmatrix}_{n\times 1} & \cdots \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

  - 2) A proper (det($A$) = 1) orthogonal matrix is a rotation matrix
    - this follows from the fact that it does not change the norms ("sizes") of the vectors on which it operates,

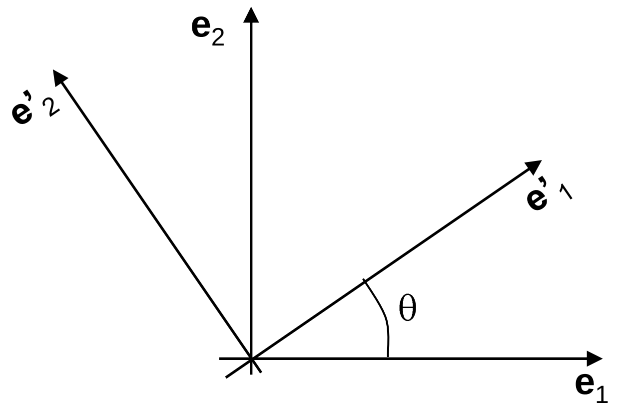
$$\|Ax\|^2 = (Ax)^T (Ax) = x^T A^T A x = x^T x = \| x \|^2,$$

and does not induce a reflection.

# Rotation matrices

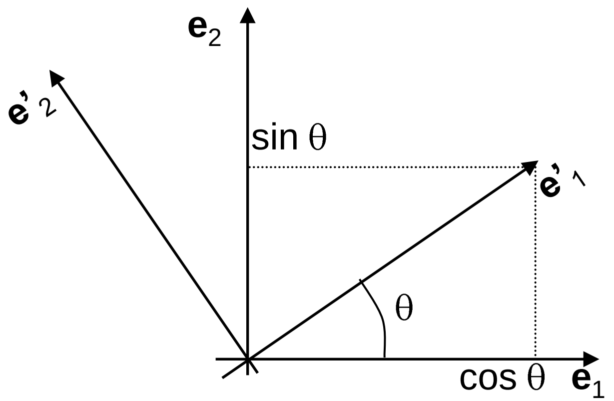
- The combination of
  1. "operator" interpretation
  2. "block matrix trick"

  is useful in many situations
- Poll:
  - "What is the matrix **R** that rotates the plane $R^2$ by $\theta$ degrees?"



# Rotation matrices

- The key is to consider how the matrix operates on the vectors $\mathbf{e}_i$ of the canonical basis
  - note that R sends $\mathbf{e}_1$ to $\mathbf{e'}_1$




$$e'_1 = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

  - using the column space picture

$$e'_1 = \begin{pmatrix} r_{11} \\ r_{21} \end{pmatrix} \times 1 + \begin{pmatrix} r_{12} \\ r_{22} \end{pmatrix} \times 0 = \begin{pmatrix} r_{11} \\ r_{21} \end{pmatrix}$$

  - from which we have the first column of the matrix

$$R = \begin{bmatrix} e'_1 & \begin{matrix} r_{12} \\ r_{22} \end{matrix} \end{bmatrix} = \begin{bmatrix} \cos\theta & r_{12} \\ \sin\theta & r_{22} \end{bmatrix}$$

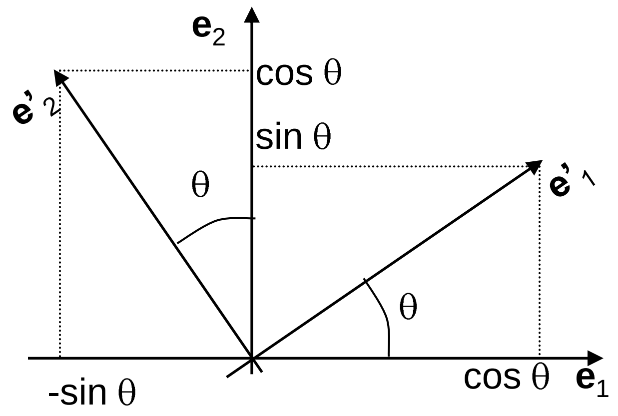
# Rotation Matrices

- and we do the same for $\mathbf{e}_2$
  - $\mathbf{R}$ sends $\mathbf{e}_2$ to $\mathbf{e'}_2$

$$e'_2 = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{pmatrix} r_{11} \\ r_{21} \end{pmatrix} \times 0 + \begin{pmatrix} r_{12} \\ r_{22} \end{pmatrix} \times 1 = \begin{pmatrix} r_{12} \\ r_{22} \end{pmatrix}$$

  - from which

$$R = [e'_1 \quad e'_2] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$




  - check

$$R^T R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = I$$

# Analysis/synthesis

- one interesting case is that of matrices with orthogonal columns
- note that, in this case, the columns of A are
  - a basis of the column space of A
  - a basis of the row space of $A^T$
- this leads to an interesting interpretation of the two pictures
  - consider the projection of x into the row space of $A^T$

$$y = A^T x$$

  - due to orthonormality, x can then be synthesized by using the column space picture

$$x' = A y$$

# Analysis/synthesis

- note that this is your most common use of basis
- let the columns of A be the basis vectors $a_i$
  - the operation $y = A^T x$ projects the vector x into the basis, e.g.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \Leftrightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

this is called the canonical basis of $R^n$

  - The vector x can then be reconstructed by computing $x' = A y$, e.g.

$$\begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} y_1 + \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} y_2 + \cdots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} y_n = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

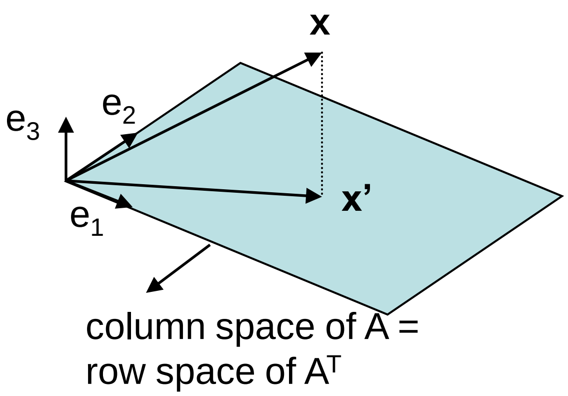
  - Q: is the synthesized x' always equal to x?

# Projections

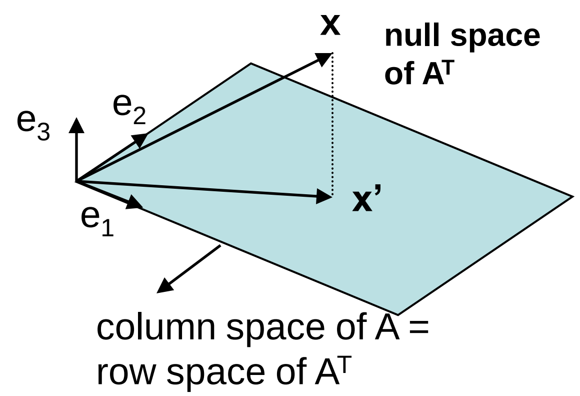
- A: not necessarily! Recall
  - $y = A^T x$ and $x' = A y$
  - $x' = x$ if and only if $AA^T = I$ !
  - this means that A has to be orthonormal.
- what happens when this is not the case?
  - we get the projection of x on the column space of A
  - e.g. let $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$ then $y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

    and

$$x' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{pmatrix} x_1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ x_2 \\ 0 \end{pmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}$$



# Null Space of a Matrix




- What happens to the part that is lost?
- This is the "null space" of $A^T$

$$N(A^T) = \{x \mid A^T x = 0\}$$

  – In the example, this is comprised of all vectors of the type $\begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix}$ since

$$A^T x = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix} = \alpha \begin{pmatrix} 0 \\ 0 \end{pmatrix} = 0$$

- FACT: *N(A)* is *always* orthogonal to the row space of *A*:
  – *x* is in the null space iff it is orthogonal to all rows of *A*
- For the previous example this means that $N(A^T)$ is orthogonal to the column space of *A*

# Orthonormal matrices

- Q: why is the orthonormal case special?
- because here there is no null space of $A^T$
- recall that for all $x$ in $N(A^T)$
  - $A^T x = 0 \Leftrightarrow x = A0 = 0$
- the only vector in the null space is 0
- this makes sense:
  - A has n orthonormal columns, e.g. $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
  - these span all of $R^n$
  - there is no extra room for an orthogonal space
  - the null space of $A^T$ has to be empty
  - the projection into row space of $A^T$ (=column space of $A$) is the vector $x$ itself
- in this case, we say that the matrix has full rank

# The **Four Fundamental Subspaces**

- These exist for any matrix:
  - Column Space: space spanned by the columns
  - Row Space: space spanned by the rows
  - Nullspace: space of vectors orthogonal to all rows (also known as the orthogonal complement of the row space)
  - Left Nullspace: space of vectors orthogonal to all columns (also known as the orthogonal complement of the column space)

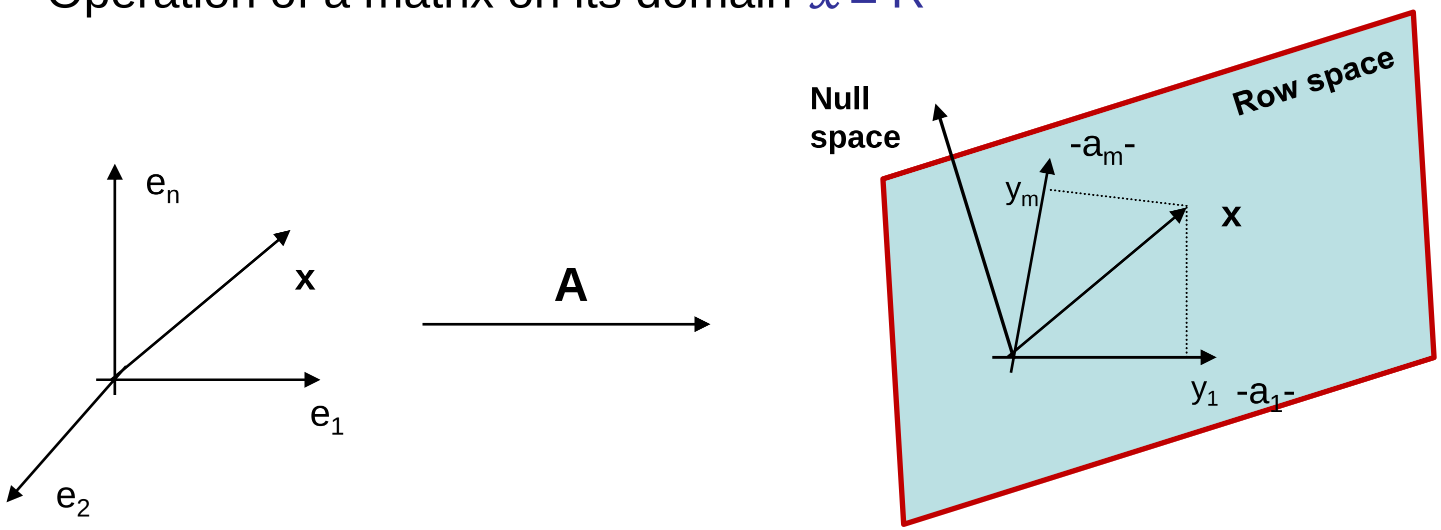
- You can think of these in the following way
  - Row and Nullspace characterize the domain space (inputs)
  - Column and Left Nullspace characterize the codomain space (outputs)

# Domain viewpoint

- Domain $\mathfrak{X} = R^n$
  - y = coordinates of x in row space of A
  - Row space: space of "useful inputs", which A maps to non-zero output
  - Null space: space of "useless inputs", mapped to zero
  - Operation of a matrix on its domain $\mathfrak{X} = R^n$

$$\begin{bmatrix} \vdots \\ y_i \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ - & a_i & - \\ & \vdots & \end{bmatrix} x \quad (m \text{ rows})$$

$$N(A) = \{x \mid Ax = 0\}$$



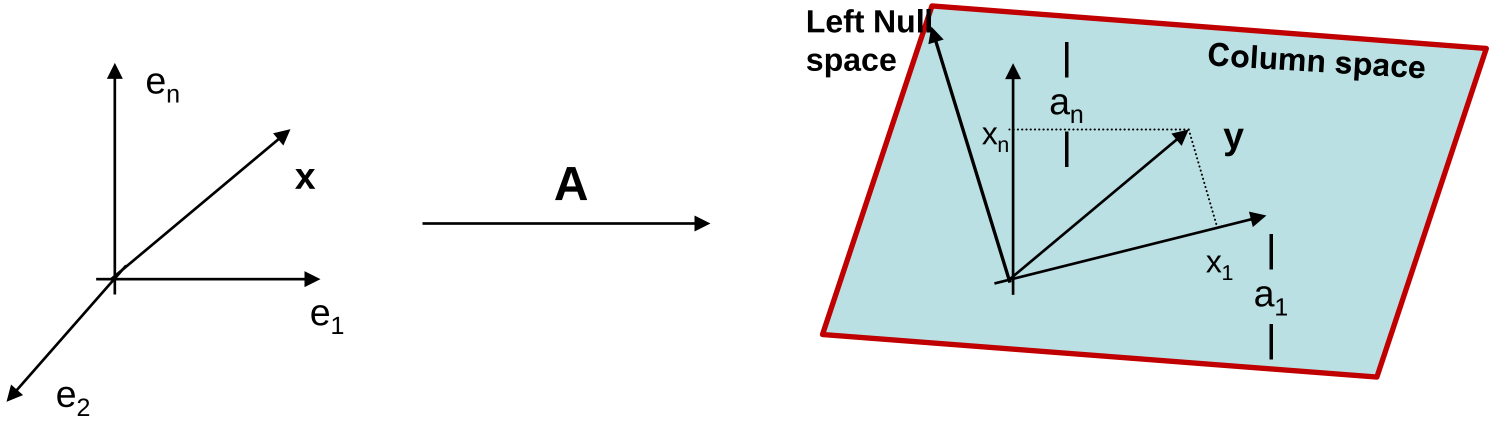

  - Q: what is the null space of a low-pass filter?

# Codomain viewpoint

- Codomain $\mathcal{Y} = R^m$
  - x = coordinates of y in column space of A
  - Column space: space of "possible outputs", which A can reach
  - Left Null space: space of "impossible outputs", cannot be reached
  - Operation of a matrix on its codomain $\mathcal{Y} = R^m$

$$y = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} x_1 + \cdots + \begin{bmatrix} | \\ a_n \\ | \end{bmatrix} x_n$$

$$L(A) = \{ y \mid y^T A = 0 \}$$




  - Q: what is the column space of a low-pass filter?

# The **Four Fundamental Subspaces**

**Assume Domain of *A* = Codomain of *A*. Then:**

- **Special Case I:** Square Symmetric Matrices ($A = A^T$):
  - Column Space is equal to the Row Space
  - Nullspace is equal to the Left Nullspace, and is therefore orthogonal to the Column Space
- **Special Case II:** nxn Orthogonal Matrices ($A^TA = AA^T = I$)
  - Column Space = Row Space = $R^n$
  - Nullspace = Left Nullspace = $\{0\}$ = the Trivial Subspace

# Linear systems as matrices

- A linear and time invariant system
  - of impulse response *h[n]*
  - responds to signal *x[n]* with output $y[n] = \sum_k x[k]h[n-k]$
  - this is the convolution of *x[n]* with *h[n]*
- The system is characterized by a matrix
  - note that

$$y[n] = \sum_k x[k] g_n[k], \quad \text{with } g_n[k] = h[n-k]$$

  - the output is the projection of the input on the space spanned by the functions $g_n[k]$

$$\begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[n] \end{bmatrix} = \begin{bmatrix} - g_1 - \\ - g_2 - \\ \vdots \\ - g_n - \end{bmatrix} x = \begin{bmatrix} h[0] & h[-1] & \cdots & h[-(n-1)] \\ h[1] & h[0] & \cdots & h[-(n-2)] \\ & & \ddots & \\ h[n-1] & h[n-2] & \cdots & h[0] \end{bmatrix} \begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[n] \end{bmatrix}$$

# Linear systems as matrices

- the matrix

$$A = \begin{bmatrix} h[0] & h[-1] & \cdots & h[-(n-1)] \\ h[1] & h[0] & \cdots & h[-(n-2)] \\ & & \ddots & \\ h[n-1] & h[n-2] & \cdots & h[0] \end{bmatrix}$$

  – characterizes the response of the system to any input
  – the system projects the input into shifted and flipped copies of its impulse response *h[n]*
  – note that the column space is the space spanned by the vectors h[n], h[n-1], …
  – this is the reason why the impulse response determines the output of the system
  – e.g. a low-pass filter is a filter such that the column space of A only contains low-pass low pass signals
  – e.g. if h[n] is the delta function, A is the identity

# Any questions?