David Pacheco

1/04

Bases are useful because we can assign real coordinates to vectors. We can represent objects as columns, which computers can easily manipulate.

Linear Transformations + Matrix Vector Multiplication.

Def. A transformation is a rule that takes an argument (input) and assigns it a value (output).

domain of transformation: set of inputs
target space of transformation: set of values
(or codomain)

$$\boxed{T : X \to Y}$$

Transformation "T" with domain X and codomain Y.
(target space).

Example: class list domain (number)
codomain (name)

Function, operator are synonyms for transformation.

Def. Let V, W be vector spaces.
$T : V \to W$ is linear if
① $T(\vec{u}+\vec{v}) = T(\vec{u}) + T(\vec{v})$
② $T(\alpha \vec{u}) = \alpha T(\vec{u})$

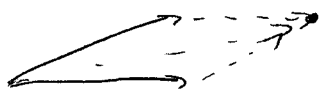
In other words: $T(\alpha\vec{u} + \beta\vec{v}) = \alpha T(\vec{u}) + \beta T(\vec{v})$

<u>amples</u>: rotating, stretching the distance of a pt from the origin. Also, <u>differentiation</u>.

(i) $V = \mathbb{P}_n$, $W = \mathbb{P}_n$ (Differentiation)

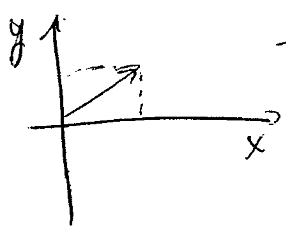
$T(f) = f'$

(2) (Rotation of x,y plane by α)



To rotate this sum (of vectors), it does not matter if you rotate and then add, or add first.

3) Reflection on x-axis



$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ -y \end{pmatrix}$

4) $T: \mathbb{R}^1 \to \mathbb{R}^1$ (multiplication by a number) for any such T.

$T(x) = ax$

why? Because

$T(x) = T(x \cdot 1) = x \cdot T(1) = ax, \quad a = T(1)$

because T is a linear transformation, $T(\alpha x) = \alpha T(x)$

Prop: Let $\vec{v}_1 \dots \vec{v}_n$ be a basis in V and let $T: V \to W$.

If we know $T(\vec{v}_1), T(\vec{v}_2) \dots T(\vec{v}_n)$, we can compute $T(\vec{v})$ $\forall \vec{v}$.
(That is, it is sufficient to know T for the basis)

Pf.
$$\vec{v} = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \dots + \alpha_n \vec{v}_n$$
$$T(\vec{v}) = T\left(\sum_{k=1}^{n} \alpha_k \vec{v}_k\right)$$
$$= \alpha_1 T(\vec{v}_1) + \alpha_2 T(\vec{v}_2) + \dots + \alpha_n T(\vec{v}_n)$$
these are known (given)

Applying this:
let $T: \mathbb{R}^n \to \mathbb{R}^m$
let $\vec{e}_1 \dots \vec{e}_n$ be the standard basis in $\mathbb{R}^n$.

$$\vec{a}_1 = T(\vec{e}_1), \quad \vec{a}_2 = T(\vec{e}_2), \dots, \quad \vec{a}_n = T(\vec{e}_n).$$
$(a_k \in \mathbb{R}^m)$

Join $\vec{a}_k$ in a matrix:
$A = [\vec{a}_1, \vec{a}_2, \dots \vec{a}_n]$ m rows, n cols (because each a_k has m rows)

Matrix multiplication is such that $T(\vec{x}) = A\vec{x}$

$$\vec{x} = \sum_{k=1}^{n} x_k \vec{e}_k$$
$$T(x) = x_1 T(\vec{e}_1) + x_2 T(\vec{e}_2) + \dots + x_n T(\vec{e}_n)$$
$$= x_1 \vec{a}_1 + x_2 \vec{a}_2 + \dots + x_n \vec{a}_n$$

Column-by-coordinate rule of matrix multiplication:

Take col. #k of A,
Multiply by coord. #k of $\vec{x}$
Add.

Example:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 1 \cdot \begin{bmatrix} 1 \\ 4 \end{bmatrix} + 2 \begin{bmatrix} 2 \\ 5 \end{bmatrix} + 3 \begin{bmatrix} 3 \\ 6 \end{bmatrix}$$

This works well for parallel computing. For us:

Row-by-column rule:

entry #k of $A\vec{x}$ = (row #k of A) $\cdot$ ($\vec{x}$)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 3 \\ 1 \cdot 4 + 5 \cdot 1 + 6 \cdot 3 \end{bmatrix}$$

Bad Example:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

NOT DEFINED - you run out of entries in the vector.