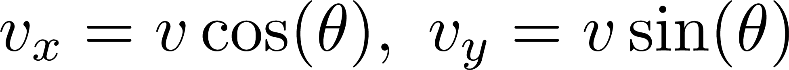
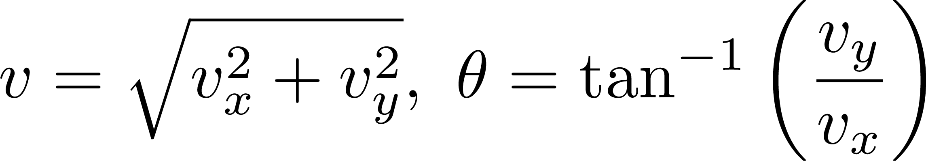
# Material Summary: Linear Algebra

## Vectors

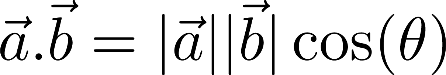
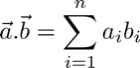
### Vector Definitions

* "Physics definition"
  + A pointed segment in space
* "Computer science definition"
  + A list of objects (usually numbers)
  + Dimensions = length
* Math definition
  + Encompasses both, and allows even more abstraction: *\_*𝑣*\_*
  + Vectors can be added and multiplied
    - By numbers and other vectors
  + Similar to how we defined a field
* Another perspective
  + Transformations
  + Actually, things a just a little more complicated…
    - You can look up "tensors" if you're interested
    - We'll talk a little about tensors later

### Vector Components

* The distances to all coordinate axes: *\_*𝑣*\_*𝑥*\_, \_*𝑣*\_*𝑦*\_*
* Equivalent to
* Polar coordinates:
* Finding components: Pythagoras
* Finding the polar form (magnitude, direction): 
* All these operations generalize to more than 2 dimensions
* We usually denote vectors by *\_*𝑣*\_* or with bold type: 𝒗
  + Another notation: Latin letters for vectors, Greek letters for numbers
  + Reason: The vector 𝒗 and its length 𝑣 can be easily confused

### Vector Operations

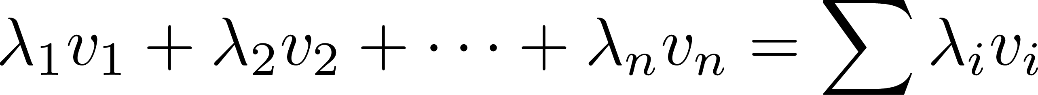
* **Addition**
  + Result:  length = distance from start to end,  direction: start → end
  + In component form: sum all components for every direction
* **Multiplication by a number (scalar)**
  + Result: length = scaled length, direction: same (if scalar *≥0*), opposite otherwise
  + In component form: multiply each component by the number
* **Scalar product of two vectors** 
  + Also called dot product or inner product
  + Result: scalar
  + Definition: 
  + Using the vector components: 
* **Vector product of two vectors**
  + Also called cross product
  + Result: vector, perpendicular to both initial vectors
  + Definition: 
    - – normal vector
    - Magnitude: = area of parallelogram   
      between and
    - Direction: coincides with the direction of

### Vector Spaces

* A field (usually R or C): 𝐹
* A set of elements (vectors): 𝑉
* Operations
* Addition of two vectors: 𝑤*=*𝑢*+*𝑣
* Multiplication by an element of the field: 𝑤*=*𝜆𝑢
* A "checklist" of eight axioms
* We read this as "vector space (or linear space) 𝑉 over the field 𝐹"
* Examples of vector spaces
  + Coordinate space, e.g., real coordinate space
    - 𝑛-dimensional vectors
  + Infinite coordinate space 
    - Vectors with infinitely many components
  + Polynomial space
    - All polynomials of variable 𝑥 with real coefficients
  + Function space

### Linear Combinations

* Vectors:
* Numbers (scalars):
* **Linear combination:** The sum of each vector multiplied by a scalar coefficient

****

* Why linear? No fancy functions, no vector multiplications
* **Span** (linear hull) of vectors: the set of all their linear combinations
* Linear (in)dependence
  + The vectors are **linearly independent** if the only solution to the equation is
  + Conversely, they are **linearly dependent** if there is a non-trivial linear combination which is equal to zero
* Example:



* A black background with a black square

  Description automatically generated with medium confidenceConsider:
* A line of arrows pointing to different directions

  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidenceNow consider the vector:
* A black background with a black square

  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidenceWe can see that we can express as the linear combination:
* **Linearly independent**
* Every other vector in the space can be represented as their linear combination
* This linear combination is **unique**
* **Each vector space has a basis**
* Each pair of **two** LI vectors forms a basis  
  in **2D** coordinate space
* Each set of  **LI vectors forms a basis  
  in -dimensional vector space**

## Matrices

### Definition

* A **rectangular table of numbers**
* Dimensions: **rows columns**
* A black background with a black square

  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidenceExamples: A black background with a black square

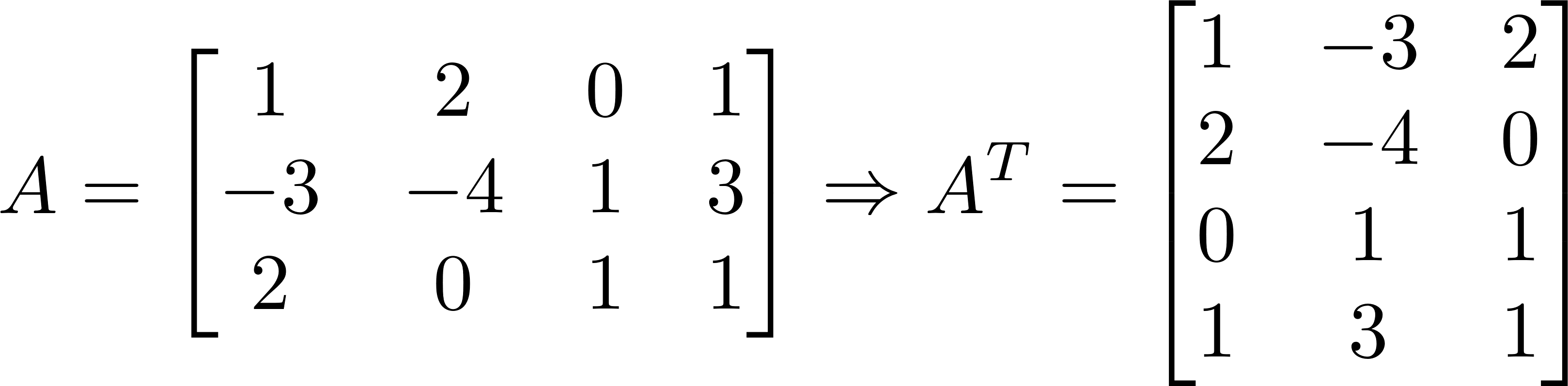
  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidence
* – row vector, – column vector
* Elements
* Scalars have **no** dimensions: 2; 3; 18; -42; 0,5
* Vectors have **one** dimension:
* Matrices have **two** dimensions:
* A generalization of this pattern to many dimensions is called a **tensor**
  + Tensors are quite more complicated than this
  + For almost all purposes it's OK to think about them  
    as multidimensional matrices

### Operation

* A black background with a black square

  Description automatically generated with medium confidence**Addition** (the dimensions must be the same)
* **Multiplication by a scalarA black background with a black square

  Description automatically generated with medium confidence**
* All matrices **form a vector space**
  + You may check this
* Transposition:
  + Turning **rows into columns** and vice versa
  + The transpose of a matrix is denoted by an **upper index T**

A black background with a black square

Description automatically generated with medium confidence

### Matrix Multiplication

* A black background with a black square

  Description automatically generated with medium confidenceThe dimensions **must match**:
* A black background with a black square

  Description automatically generated with medium confidenceDefinition:
* A black background with a black square

  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidenceExample:

A black background with a black square

Description automatically generated with medium confidence

* Note that
  + In this case, we can't even **multiply**
  + We say that **matrix multiplication  
    is not commutative**
    - Compare with numbers:

A screenshot of a computer

Description automatically generated **5.3 = 3.5 commutative**

* We can use @ or dot() for both matrix multiplicationand dot products
* A screenshot of a computer program

  Description automatically generated**Note:** Whenever possible, use **numpy** arrays instead of lists

### Transformation

* A mapping (function) between two vector spaces:
* Special case: mapping a space onto itself:
  + This is called a linear operator
* A grid with a red and blue rectangle

  Description automatically generatedEach vector of gets mapped to a vector in

### Linear Transformations

* Only **linear combinations are allowed**
* The origin **remains fixed**
* All lines remain lines (not curves)
* All lines remain evenly spaced (equidistant)
* Each **space has a basis**
  + All other vectors can be expressed as **linear combinations of the basis vectors**
  + If we know how **basis vectors are transformed**, we can transform **every other vector**
* A black background with a black square

  Description automatically generated with medium confidenceA black background with a black square

  Description automatically generated with medium confidenceConsider the transformation
* Consider another vector
  + A black background with a black square

    Description automatically generated with medium confidenceA black background with a black square

    Description automatically generated with medium confidenceOld basis:
  + A black background with a black square

    Description automatically generated with medium confidenceA black background with a black square

    Description automatically generated with medium confidenceNew basis:
    - **Same coefficients, new basis vectors**
* A diagram of shear with red and blue lines

  Description automatically generatedThis operation is called applying the linear transformation

### Multiple Transformations

* A black background with a black square

  Description automatically generated with medium confidenceConsider the same transformation:
* We applied the linear transformation by taking dot products
  + A black background with a black square

    Description automatically generated with medium confidenceTherefore, we can describe it in another way – using a matrix
  + This is called the **matrix of the linear transformation**
  + Its **columns** denote where the **basis vectors** go
* A black background with a black square

  Description automatically generated with medium confidenceApplying the transformation to a vector is the same as multiplying the matrix times the original vector:
* Example:

A black background with a black square

Description automatically generated with medium confidenceA black background with a black square

Description automatically generated with medium confidence

* We can apply many transformations, one right after the other
  + Result: composite transformation
  + We do this by multiplying **on the left** by the matrix of each transformation

matrix multiplication applying many transformations

* To visualize transformations, you can use the code in the **visualize\_transformation.py** file
* Intuition
  + Apply each transformation in order
  + After the last one, record where the basis vectors land
  + The new matrix is the matrix of the composite transformation
* We can either apply all transformations one by one
  + Or **just** the **resulting** transformation ☺
* This is especially useful in computer graphics
* Rotation, then shearing:

A black background with a black square

Description automatically generated with medium confidence

* A black background with a black square

  Description automatically generated with medium confidenceApply rotation to a vector:
* Apply shear to the resulting vector:

A black background with a black square

Description automatically generated with medium confidence

* This is the same as:

A black background with a black square

Description automatically generated with medium confidence

* A black background with a black square

  Description automatically generated with medium confidenceThe new transformation matrix is:
* Measure of how much the unit area (volume) changes
* Scalar value
* Defined only for square matrices
* For more than two dimensions: area volume
* The determinant of a matrix is denoted
* The determinant has very useful [properties](https://www.math.drexel.edu/~jwd25/LM_SPRING_07/lectures/lecture4B.html)
  + \documentclass{article}
    \usepackage{amsmath}
    \pagestyle{empty}
    \begin{document}

    $$ \det(AB) = \det(A) \det(B) $$


    \end{document}Notably,

## Linear Systems

### Linear Systems in Matrix Form

* Consider the linear system

A black background with a black square

Description automatically generated with medium confidence

* Unknown variables
* We can represent this as a matrix equation

A black background with a black square

Description automatically generated with medium confidence

* + Or more generally:
  + Looks like a linear equation "on steroids"

### Inverse Matrix

* Consider a **general**, "good" transformation
  + The inverse transformation will **"bring back" the basis vectors**
  + **clockwise rotation counterclockwise rotation**
* The inverse transformation has its own matrix:
* If we apply the transformation and the inverse,  
  we'll get our initial result
  + I.e., nothing will change
  + In math terms:
* Let's now try to apply the inverse transformation  
  to our linear system
  + Note that this means multiplying on the left

A black background with a black square

Description automatically generated with medium confidenceA black background with a black square

Description automatically generated with medium confidence

* To find the **unknown vector** 
  + We need to find the **inverse matrix of**
  + There are many methods, the most popular of which   
    is called **Gaussian elimination** (or Gauss – Jordan method)
* **Basic idea**:
  + Apply some transformation to get **from to**
  + Apply the **same transformation to**
  + What we get is **the inverse matrix**