1. Basic Vectors

A vector is a mathematical entity that has both magnitude and direction. Vectors are fundamental in physics, engineering, and computer science.

1.1 Vector Representation
- A vector in a 2D plane is written as A = (x, y), where:
  - x is the horizontal component. - y is the vertical component.
- In 3D, a vector is A = (x, y, z).

1.2 Vector Types

Unit vector is a vector with a magnitude of 1 with a specified direction. If A were a vector, then:

A = [3, 4] (Along the *x* axis and *y* axis)

Axis vector represents the direction of a specific axis, defining the orientation. In a 3D coordinate system, axis vectors would correspond to *x, y, z* axes.

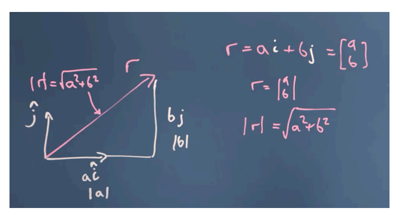
The **resultant vector** is the sum of two or more vectors. If A and B are vectors, then:

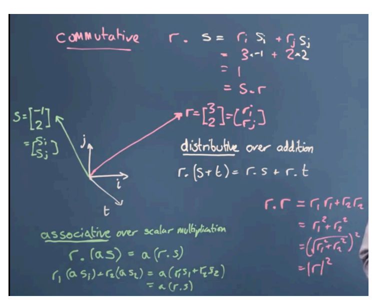
R = A + B

Using component-wise addition:

$R_x = A_x + B_x, R_y = A_y + B_y$

1.3 Dot Product





The dot product of two vectors A and B is given by:

$A \cdot B = |A|\,|B| \cos(\theta)$

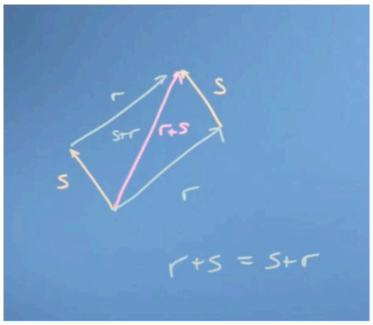
Where:
- |A| and |B| are magnitudes of vectors A and B.
- θ is the angle between them.

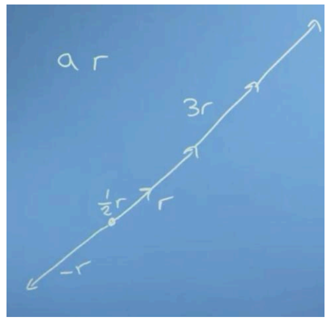
Component form:

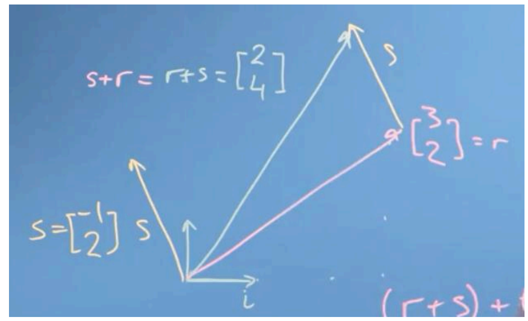
$A \cdot B = A_x B_x + A_y B_y + A_z B_z$

1.4 Vector Operations
- Addition: **$A + B = (A_x + B_x, A_y + B_y)$**
- Subtraction: **$A - B = (A_x - B_x, A_y - B_y)$**

- Scalar Multiplication: **kA = (kA_x, kA_y)**
- Magnitude: |**A**| = √**(A_x² + A_y²)**
- Cross Product:
  A × B = (A_yB_z - A_zB_y, A_zB_x - A_xB_z, A_xB_y - A_yB_x)





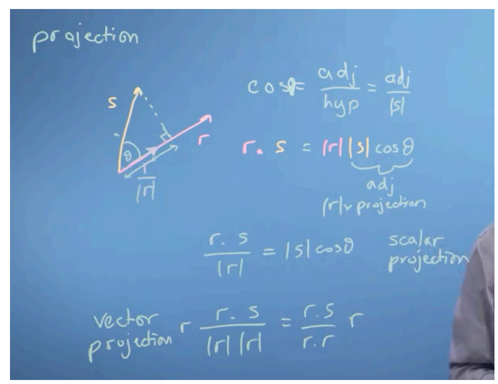


2. Projection of Vectors

The projection of vector A onto vector B (denoted as Proj_B(A)) is given by:

Proj_B(A) = (A · B) / |B|² B

This represents how much of vector A lies in the direction of B.



Scalar Projection:

Scalar Projection = (A · B) / |B|

Finding Coordinates via Projection:
If B1 and B2 are basis vectors, the coordinates of A in this basis are:

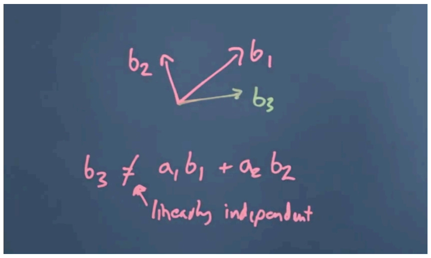
c_1 = (A · B1) / |B1|², c_2 = (A · B2) / |B2|²

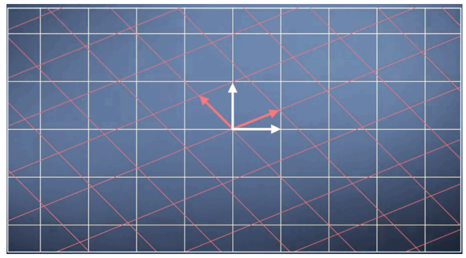
3. Change of Basis

Basis: Set of *n* vectors that are not linear combinations of each other (**linearly independent;** combinations of basis vectors cannot result in the alternative basis)

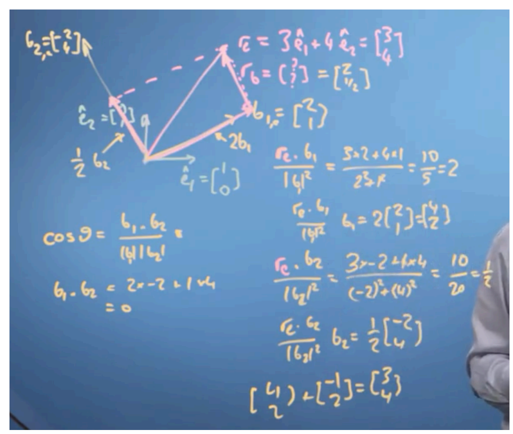
A vector may be expressed in different bases. Suppose you have:

- An original basis {e_1, e_2}
- A new basis {b_1, b_2}





3.1 Methods for Change of Basis



Direct Decomposition Method (for Orthogonal Basis):
If b_1 and b_2 are orthogonal to e_1 and e_2:
- e_1 = (R · b_1) / |b_1|²

- e_2 = (R · b_2) / |b_2|²

Change-of-Basis Matrix Method (for General Basis):
1. Construct the Basis Matrix:
   B = |b_1 b_2|
2. Compute B^{-1}.
3. Transform the coordinates:
   $R' = B^{-1} R$

Example:
For R = (3,4) and new basis B1 = (1,1), B2 = (1,-1):

c_1 = (3,4) · (1,1) / (1² + 1²) = 7 / 2 = 3.5
c_2 = (3,4) · (1,-1) / (1² + (-1)²) = -1 / 2 = -0.5

Thus, R' = (3.5, -0.5) on the new basis.

4. Properties of Vector Operations

4.1 Vector Addition Properties

a. Commutative Property
A + B = B + A

b. Associative Property
(A + B) + C = A + (B + C)

c. Additive Identity
A + 0 = A

d. Additive Inverse
A + (-A) = 0

4.2 Vector Cross Product Properties

a. Anticommutative Property
A × B = - (B × A)

b. Distributive Property
A × (B + C) = (A × B) + (A × C)

c. Associativity with Scalars
(kA) × B = k (A × B)

5. Matrix

5.1 Matrix Multiplication

Associative Property

(AB)C = A(BC).

For any three matrices A, B, and C (with compatible dimensions for the products), the product is associative:

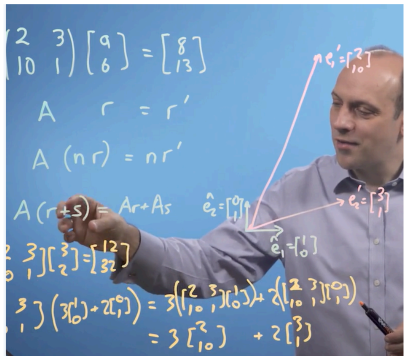
Non-Commutative Property

A.B != B.A

5.2. Matrix Space Transformation and Common Matrix Types

Matrix Space Transformation

Matrices often represent linear transformations from one vector space to another (or from a space to itself). In this context, the product of matrices corresponds to the composition of these transformations.



Matrix Transpose

The transpose of a matrix A, denoted as $\mathbf{A}^T$, is obtained by swapping its rows and columns.

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$A^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

Identity Matrix (I)

A square matrix with ones on the diagonal and zeros elsewhere:

$$I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}.$$

Allows for any matrix A of compatible size to: $\mathbf{A} \cdot \mathbf{I} = \mathbf{I} \cdot \mathbf{A} = \mathbf{A}$

Inverse Matrix (A^{-1})

A square matrix A is invertible if there exists a matrix A^{-1} such that $\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{A}^{-1} \cdot \mathbf{A} = \mathbf{I}$. Helps to **undo** a linear transformation.

Rotation Matrix

A 2D rotation matrix that rotates vectors counterclockwise by an angle θ:

$R(\theta) = [[\cos\theta, -\sin\theta],$
$[\sin\theta, \cos\theta]].$

Such matrices are orthogonal; thus their inverse are equal to their transpose:

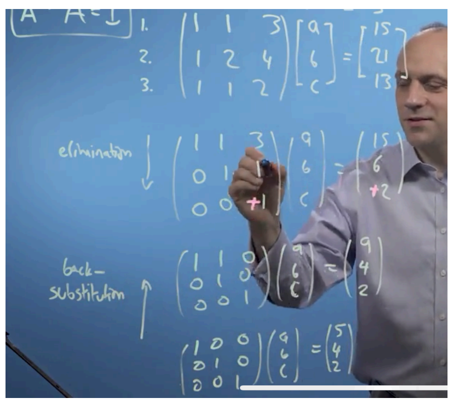
$R(\theta)^{-1} = R(\theta)^T$

Shearing Matrix
A shearing transformation in the horizontal direction:
S(m) = [[1, m]
[0, 1]]
where *m* is the shear factor. This matrix “pushes” one coordinate in proportion to the other, tilting the shape without changing its area (in 2D).

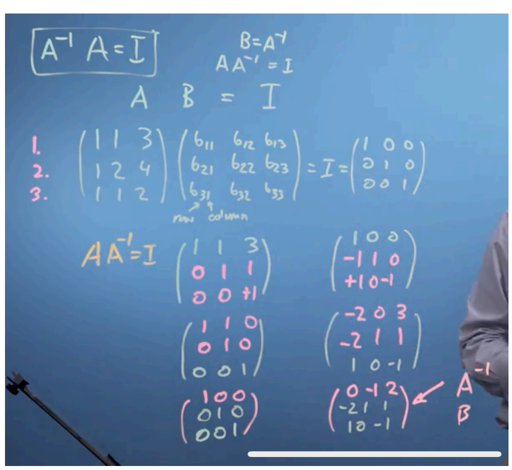
5.3 Gaussian Elimination and Triangular Matrices

Gaussian elimination is a systematic method for solving systems of linear equations without computing the inverse of the coefficient *m*.

Process Overview:
1. Use elementary row operations (swapping rows, multiplying a row by a nonzero scalar, and adding a multiple of one row to another) to transform the augmented matrix of a system into an upper triangular (or row-echelon) form
2. The resulting matrix will be upper triangular (all entries below the main diagonal are zeros). This simplifies solving the system.
3. Starting from the last equation (which contains only one unknown), solve for that variable and then substitute back into the previous equations to find all unknowns.



Finding the inverse using gaussian elimination:



6. Determinant and Invertibility

• The determinant of a square matrix is a scalar that summarizes certain properties of the matrix.

• If **det(A) = 0,** the matrix A is singular (non-invertible). [There is no matrix A^{-1} such that

$A \cdot A^{-1} = I$.]

• A zero determinant implies that the system of equations represented by the matrix has either no unique solution or infinitely many solutions.

• If the rows (or columns) of a matrix are linearly dependent, $det(A) = 0$.

Formula for Determinants:

- **2×2 Matrix:**

  For a matrix:

  $$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

  The determinant is calculated as:

  $$\det(A) = ad - bc$$

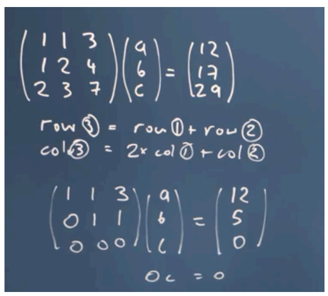
- **3×3 Matrix:**

  For a matrix:

  $$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

  The determinant is:

  $$\det(A) = a(ei - fh) - b(di - fg) + c(dh - eg)$$



7. Changing Vectors to a Different Basis Using Matrices

A vector can be expressed in different coordinate systems (bases).When the two bases are orthogonal, converting the vector from one basis to the other can be done using projections.

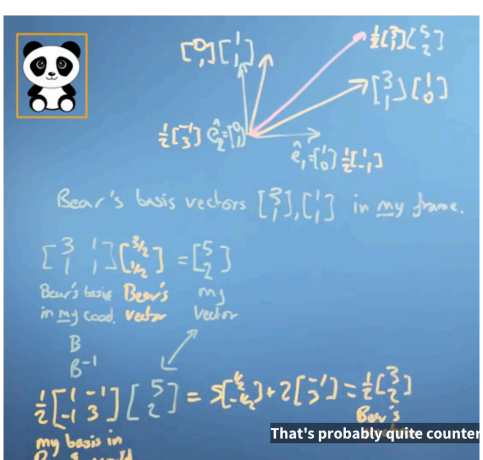
Orthogonal Bases:

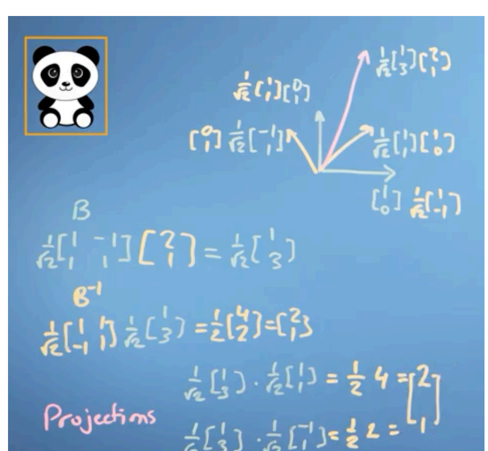
If the new basis {B1, B2} is orthogonal, the coordinate along each basis vector is given by the projection formula:

$$c_1 = \frac{\mathbf{R} \cdot \mathbf{B}_1}{\|\mathbf{B}_1\|^2}, \quad c_2 = \frac{\mathbf{R} \cdot \mathbf{B}_2}{\|\mathbf{B}_2\|^2}.$$

Example:

Imagine an image where the blue original basis is overlaid by a yellow orthogonal basis. The dot product of the vector with the unit vectors in the yellow basis yields the new coordinates.





Changing Basis via Matrices:

When converting from one basis to another, we form a change-of-basis matrix B whose columns are the new basis vectors:

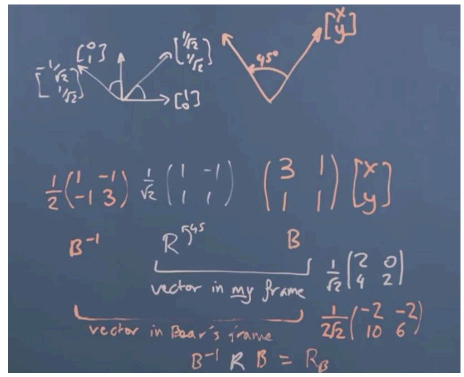
$R' = B^{-1} R$

Example:

For example, when converting the representation of a rotation, the new rotation matrix $\mathbf{R_B}$ in the new basis is given by:

$\mathbf{R_B = B^{-1}RB}$

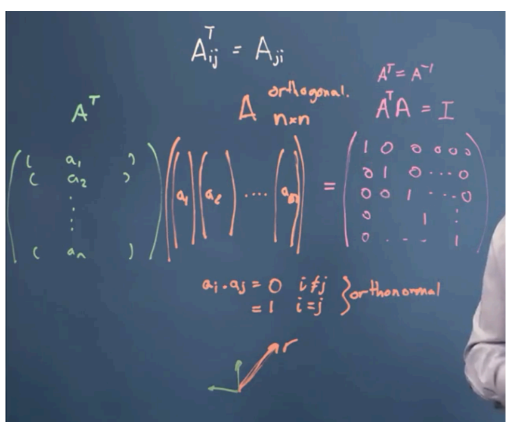
If the original basis is rotated by 45° (or any angle) to form a new basis, you can compute the rotation matrix $R(\theta)$ (with $\theta = 45^\circ$) and then convert the coordinates accordingly.



8. Orthogonal Matrices and Orthonormal Bases

An **orthonormal** basis consists of vectors that are both **orthogonal** and of unit length. allowing for:

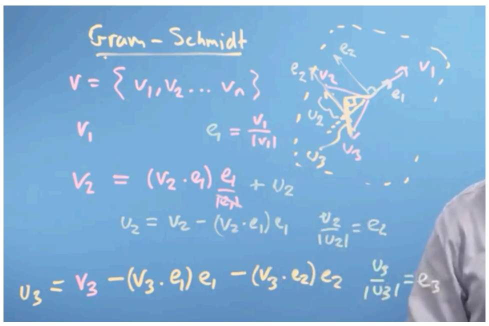
- The projection of a vector onto a basis vector is directly given by the dot product.
- For an **orthogonal** matrix Q, the inverse is simply its transpose: $Q^{-1} = Q^T$.
- With an **orthonormal** basis, the transformation matrix is orthogonal.

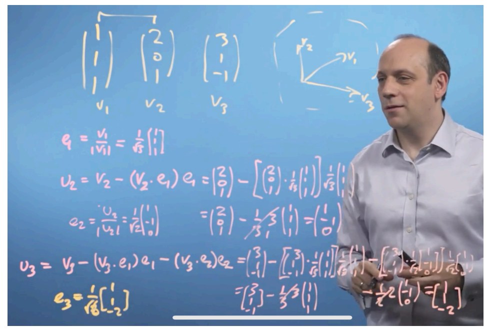


Gram–Schmidt process is used to construct an orthonormal basis from linearly independent vectors.

Summary of Process:

1. Start with a set of linearly independent vectors.
2. Subtract the projections onto the previously determined orthogonal vectors.
3. Normalize each resulting vector to obtain unit vectors.





9. Using a Basis for Vector Transformation

Any vector transformation (rotation, shear, scaling) is easier when vectors are represented in a convenient basis.

Transforming a vector to a new basis, performing the operation, and then converting back can simplify computations.

9.1 Mathematical Context

Transformation with a Basis:
Suppose you want to transform a vector ***r*** from one coordinate representation to another using a new basis. Let:

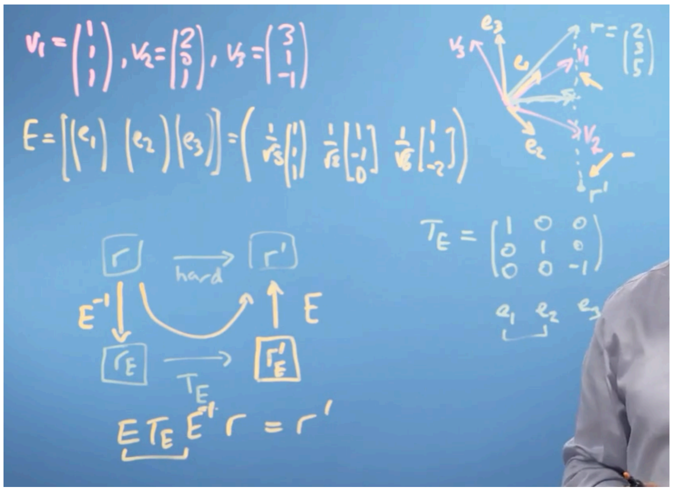
- ***E*** be a matrix whose columns are the new 3-dimensional basis vectors (obtained, for example, from original basis vectors ***v1, v2, v3***).
- T_E is the transformation matrix expressed in the new basis.
- ***E'*** denotes the inverse (or transpose if ***E*** is orthonormal) of ***E***.

Associated Formula:

The transformation of a vector r can be written as:

$\boldsymbol{r = ET_E E^{-1} r}$

(If $\boldsymbol{E}$ is orthonormal, $\boldsymbol{E^{-1}} = \boldsymbol{E^T}$, and the formula becomes $\boldsymbol{r = ET_E E^T r}$)



Interpretation:
This formula means that you first convert $\boldsymbol{r}$ into the new basis (using $\boldsymbol{E^{-1}}$), apply the transformation $\boldsymbol{T_E}$ in that basis, and then convert back to the original coordinate system with $\boldsymbol{E}$.

10. Eigenvalues and Eigenvectors

Eigenvector

Vectors whose direction (or span) remains unchanged (up to scaling) by a linear transformation. Formally, for a transformation matrix A,

$\boldsymbol{Ax = \lambda x}$**,**

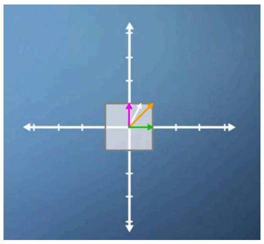
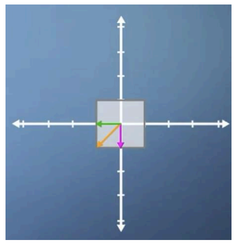
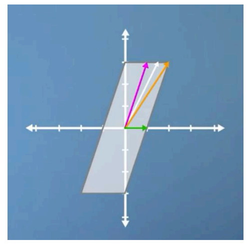
where x is an **eigenvector**.

Eigenvalues**:**
The scalar λ in the above equation, representing the factor by which the eigenvector is scaled.

Geometric Interpretation:

Under transformations such as rotations or shears, most vectors will change direction. However, eigenvectors are “special” in that they maintain their span.

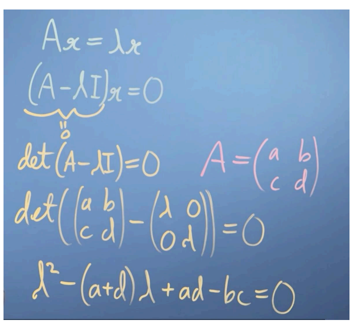
Example Context:
Several vectors (orange, green, purple) remain on their original line (despite a 180° rotation) and thus qualify as eigenvectors. In a shearing transformation, perhaps only a subset (white and green) remain as eigenvectors.

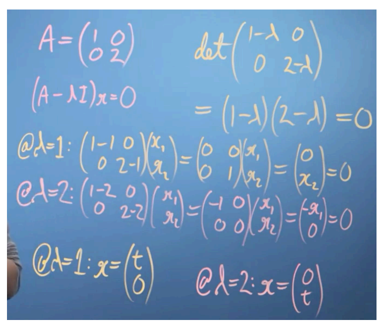
  

Computing Eigenvalues:

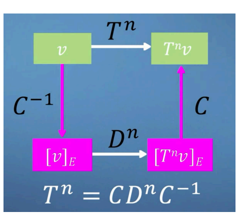
Given a transformation matrix A, eigenvalues are computed by solving the characteristic equation:

$det(A - \lambda I) = 0$.





Eigenbasis**:**
If a matrix T can be diagonalized, it is transformed into an eigenbasis. In this diagonal form, raising T to a power n becomes straightforward because diagonal matrices are easier to exponentiate.



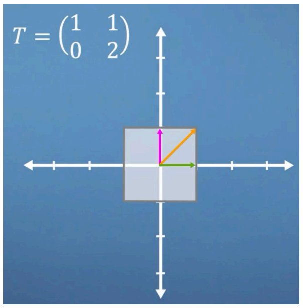
Diagonalization**:**
Transforming T into a diagonal matrix via a similarity transformation (using its eigenvectors as a basis) simplifies many computations, especially those involving matrix powers, as:

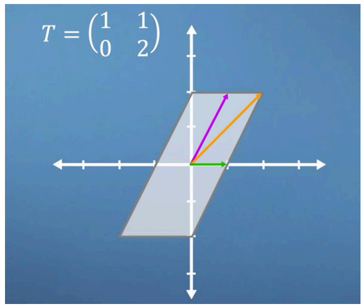
$$T^n = C\,D^n\,C^{-1}.$$

D is diagonal matrix of T

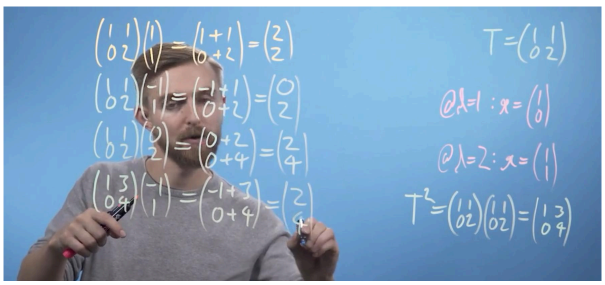
C is the matrix composed of eigenvectors

Example:

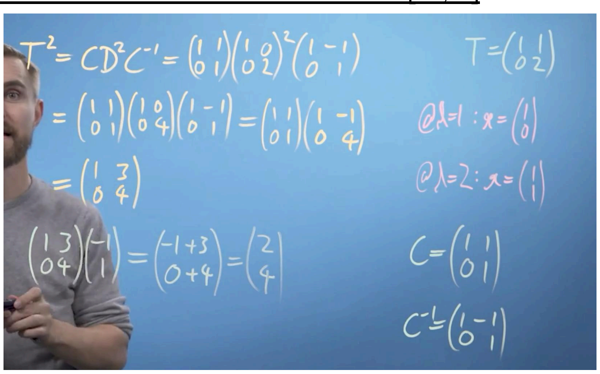




<u>Manual transformation of vector [-1, 1]</u>

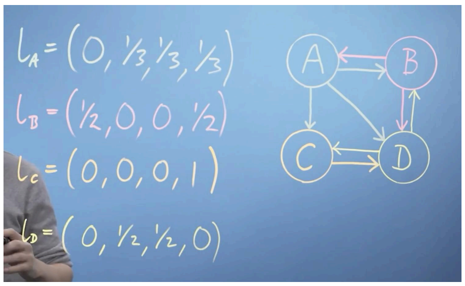


<u>Forumlic transformation of vector [-1, 1]</u>



10. PageRank

PageRank is based on links between websites. Nodes ***A, B, C, and D*** represent websites and their redirects. We can map these redirects as ***1s*** (redirecting to a page) and ***0s*** (no redirect), forming an adjacency matrix. This matrix can be used to create vectors that are then normalized, ensuring the sum of probabilities equals 1. This normalized value represents the likelihood or importance of each website in the network.



Vector *r* represents the ranks for each webpage, initially we start with equal ranks and iteratively change them until the changes are too small to be useful