Some linear algebra...

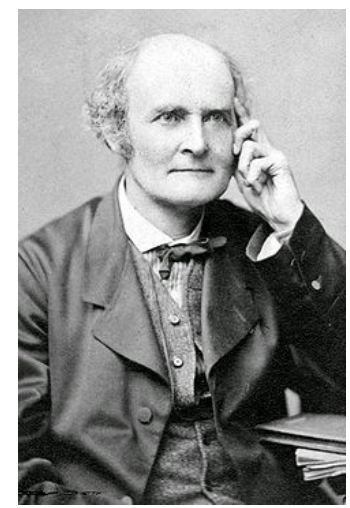
Thomas Hamelryck
Bioinformatics Center
University of Copenhagen

What is a matrix?

- A set of numbers, organized in rows and columns
  - Cayley & Silvester
    - UK, 19^{th} century (but the concept is older)
    - Work on the side by two clerks
  - Latin: *matrix;* womb
- A good thing to have

$$\begin{array}{cc} & \text{rows} \\ \text{columns} & \begin{bmatrix} a & b \\ c & d \end{bmatrix} \end{array}$$

JJ. Sylvester
(1814-1897)



A. Cayley
(1821-1895)

Addition and subtraction

- Addition
  - Easy
- Subtraction
  - Also easy

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix}$$

Multiplication I

- Row of matrix 1 times column of matrix 2

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e \\ g \end{bmatrix} = \begin{bmatrix} ae+bg \\ ce+dg \end{bmatrix}$$

- NOT commutative
  - Order matters!

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & \dots \\ \dots & \dots \end{bmatrix}$$

$$\begin{bmatrix} e & f \\ g & h \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ae+fc & \dots \\ \dots & \dots \end{bmatrix}$$

Multiplication II

- Shapes must fit
  - $\{n,m\}\times\{m,k\}\rightarrow\{n,k\}$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}\begin{bmatrix} d & e & f \end{bmatrix}=\begin{bmatrix} ad & ae & af \\ bd & be & bf \\ cd & ce & cf \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \end{bmatrix}\begin{bmatrix} d \\ e \\ f \end{bmatrix}=ad+be+cf$$

Identity matrix

- Identity element
  - {n,n} square matrix
  - $AI_0=I_0A=A$
  - Inverse A^{-1} of matrix A: $AA^{-1}=I_0$

$$I_0=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

- Matrix with ones along its diagonal, zeros elsewhere
  - $I_0=\mathrm{diag}(1,\ldots 1)$
- $\mathrm{Det}(I_0)=1$

Transpose

- Re-arrange columns and rows
  - Properties
    - $(AB)^T = B^T A^T$
    - $(A+B)^T = A^T + B^T$
    - $(A^T)^{-1} = (A^{-1})^T$
    - $Det(A^T) = Det(A)$

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

$$A = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$$

Trace

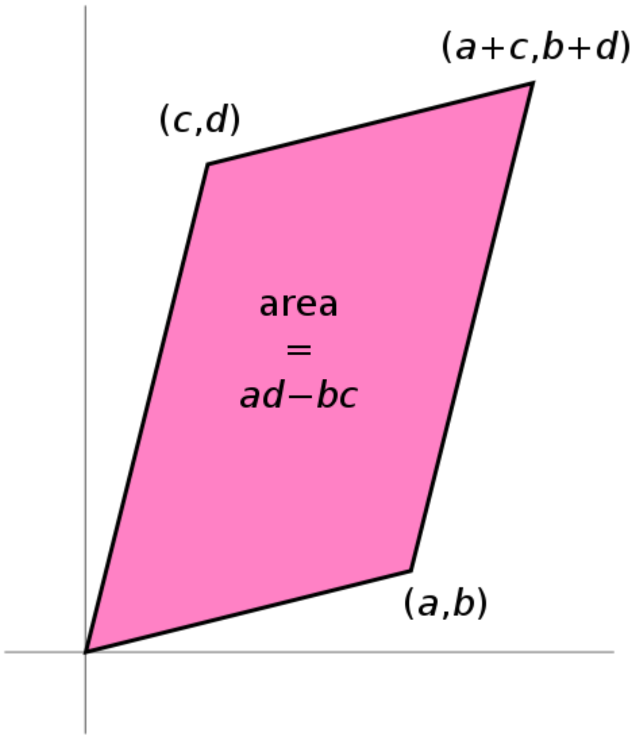
- The trace of a matrix is simply the sum of its diagonal elements
  - Defined for square matrices

$$\mathrm{Tr}\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = a+e+i$$

  - Properties:
    - Tr(A+B)=Tr(A)+Tr(B)
    - $\mathrm{Tr}(A^T)=\mathrm{Tr}(A)$
    - Tr(cA)=cTr(A)
    - Tr(AB)=Tr(BA)

Determinant

- Real number associated with a square matrix
- Volume/area enclosed by the rows, if interpreted as coordinates
- Scale factor if the matrix is interpreted as a linear transformation
  - Det(A)=0: not invertible, singular
  - Det(A)=1: rotation
  - Det(A)=-1: reflection



$$\mathrm{Det}\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

Orthogonal matrices

- Square {n,n} matrix with the following (equivalent) properties:
  - $|Det(U)|=1$
  - $U^T=U^{-1}$ and thus $UU^T=I_0$
  - The columns AND rows form an orthonormal basis of R^n
    - Vectors of length 1, mutually perpendicular
- Rotations
  - $Det(U)=1$
- Roto-reflections
  - $Det(U)=-1$

Inverse

- For invertible, square matrices A and B:
  - if $AB=BA=I_0$ then $B=A^{-1}$
- If Det(A)=0, A is singular, and does not have an inverse
- For example, for a 2-dimensional square matrix:

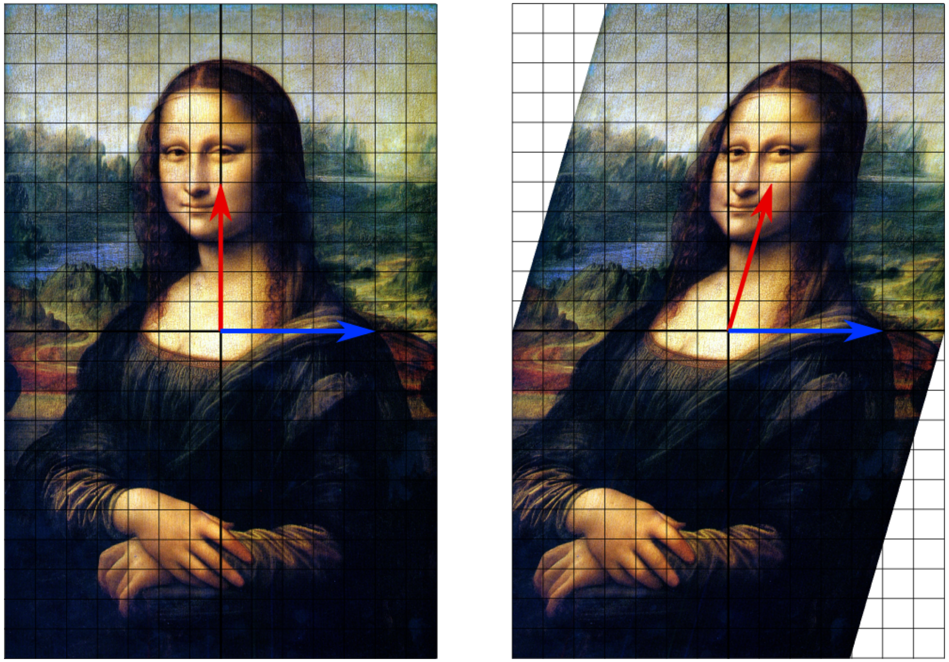
$$A=\begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow A^{-1}=\frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Eigenvalues and eigenvectors

- An eigenvector of a square matrix A is a vector **v** that does not change its direction under A's linear transformation

$$A\boldsymbol{v}=\lambda\boldsymbol{v}$$

- The scalar λ is called the eigenvalue of eigenvector **v**



The blue vector is an eigenvector with eigenvalue 1 of the shear mapping on the right.

Singular value decomposition

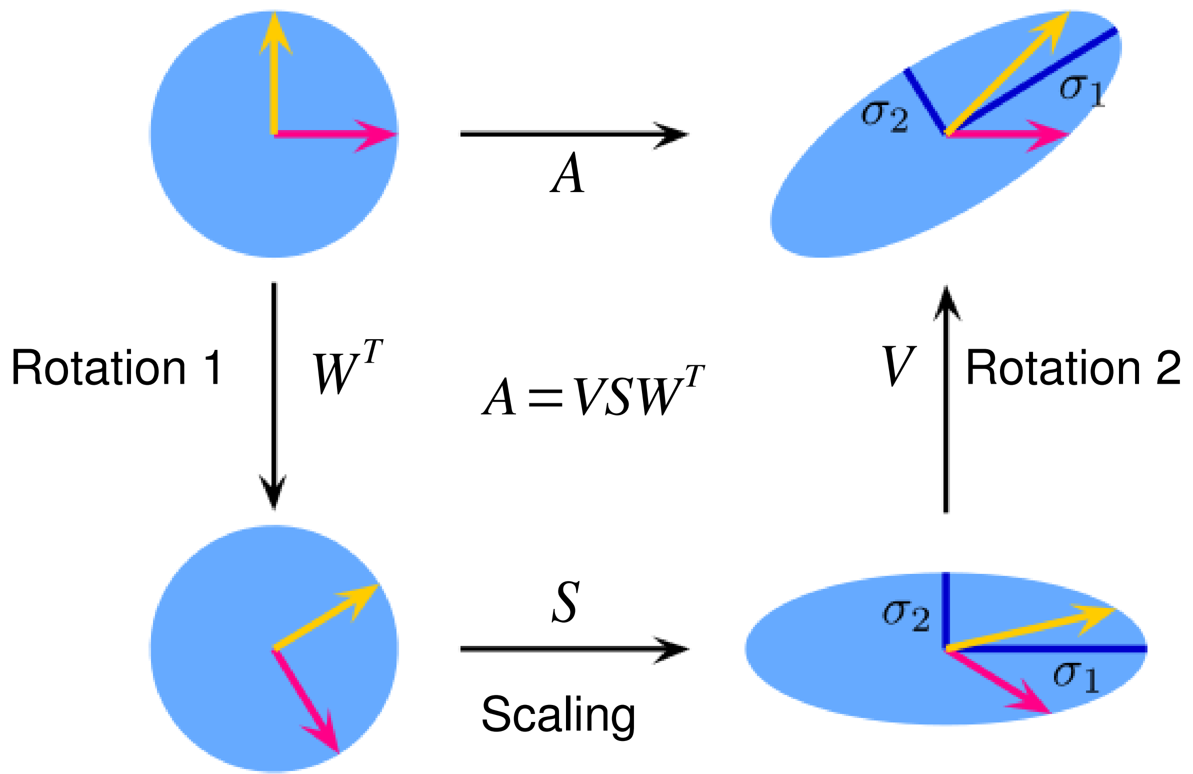
- Singular Value Decomposition theorem
  - Any real {n,m} matrix A can be written as:

$$A = VSW^T$$

  - V=orthogonal {n,n} matrix, W^T orthogonal {m,m} matrix
    - Use of transpose for W has something to do with complex matrices
      - Just a convention for us
  - S=(rectangular) diagonal {n,m} matrix
    - Non-negative values along diagonal
    - Diagonal elements are called the singular values
- Used in calculation of optimal RMSD

Singular value decomposition

- Intuitive interpretation
  - Suppose that A is a {2,2} shear matrix



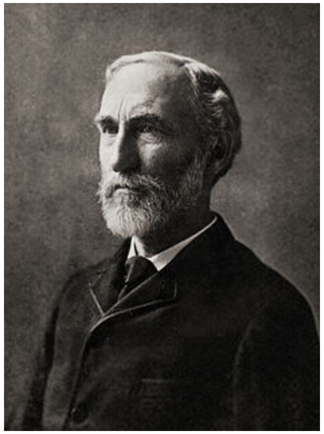
σ_1 and σ_2=Singular values

Vectors

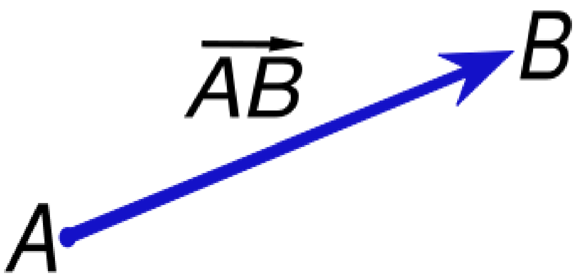
- Geometric interpretation
  - Line from point A to point B
    - with an associated length
    - with an associated direction
  - Latin: *vector;* one who carries

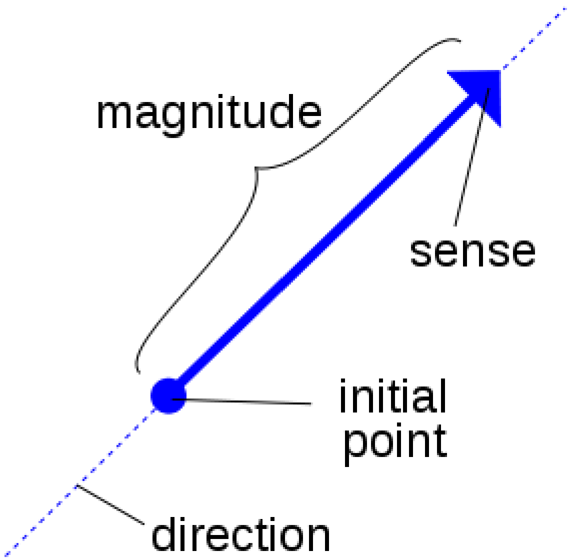
William Hamilton
(1805-1865)
Quaternions

Hermann Grassmann
(1809-1877)



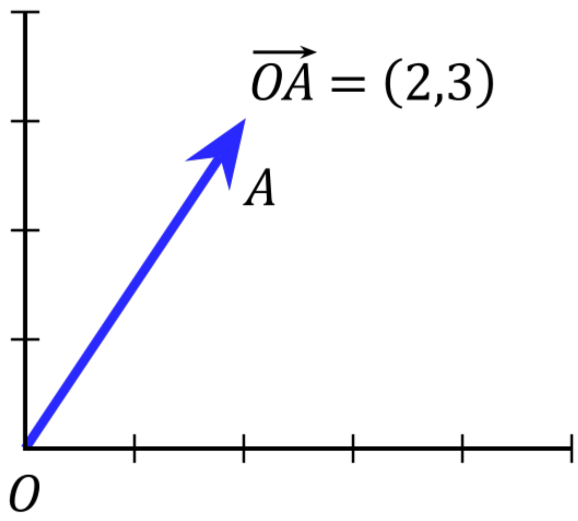
Willard Gibbs
(1839-1903)





Free vector

- For our purposes, the first point is the origin
- A vector is a column matrix
  - shape: {n,1}

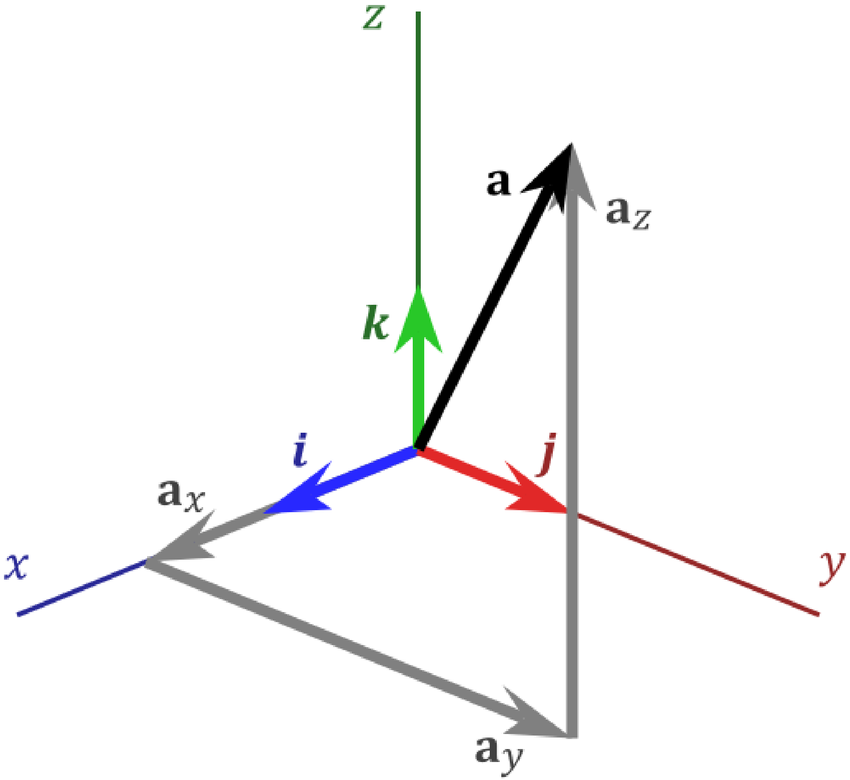


$$\vec{a} = \vec{oa} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

3D vectors

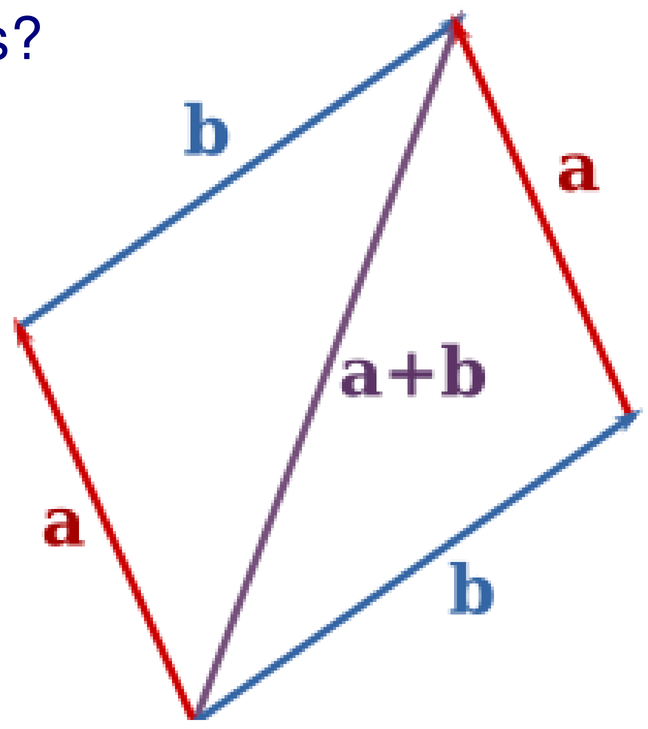
- Vectors can have any dimension

$$\vec{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$$



Vector addition and subtraction

- Just like adding/subtracting matrices
- Visualization of addition and subtraction
  - Combine head-to-tail
- Q: where are the side chains?



Dot product I

- Two vectors→real number
- Can be seen as simple matrix multiplication
- Used to calculate the norm, length or magnitude of a vector

$$\vec{a}\cdot\vec{b}=A^T B=\begin{bmatrix}a_x & a_y & a_z\end{bmatrix}\begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}=a_x b_x+a_y b_y+a_z b_z$$

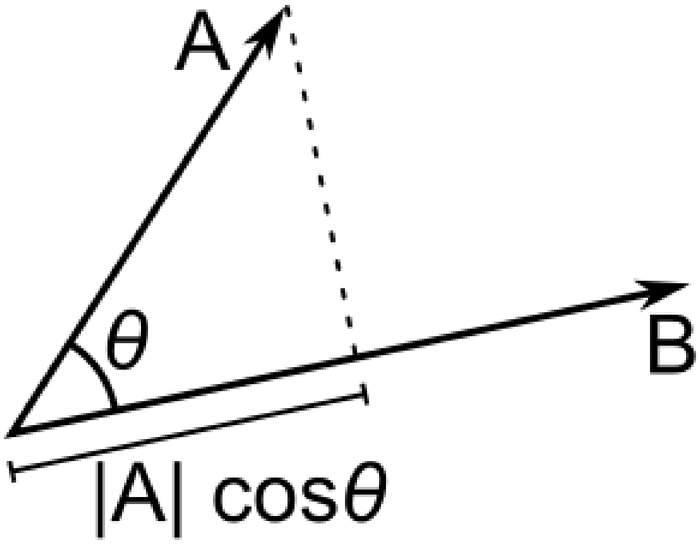
$$|\vec{a}|=\sqrt{\vec{a}\cdot\vec{a}}=\sqrt{A^T A}=\sqrt{a_x{}^2+a_y{}^2+a_z{}^2}$$

The vector's magnitude, length or norm is the square root of the dot product of a vector with itself

Dot product II

- The dot product also has a geometrical interpretation
  - Related to the angle between the vectors

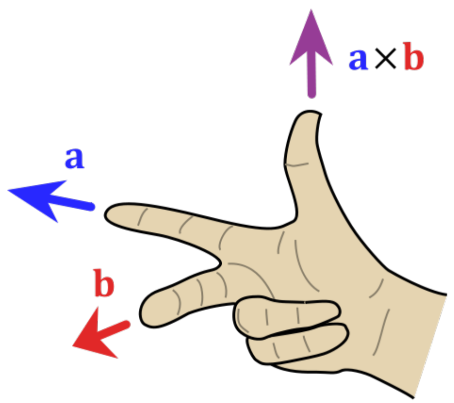
$$\vec{a} \cdot \vec{b} = |a||b| \cos(\theta)$$

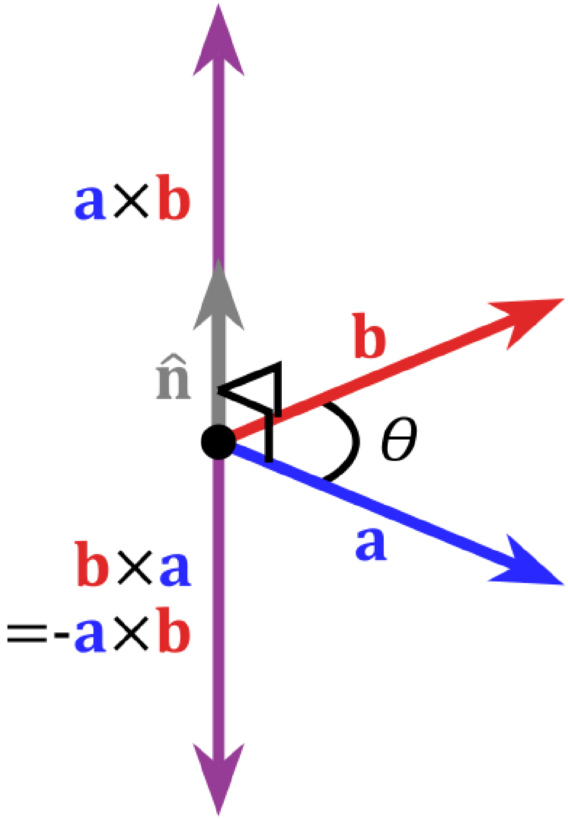


Cross product

- Cross product of two vectors is another vector
  - Perpendicular to the plane of the two vectors
    - Lengths, times sine angle, times perpendicular unit vector

$$\vec{a} \times \vec{b} = |a||b| \sin(\theta) \hat{n}$$



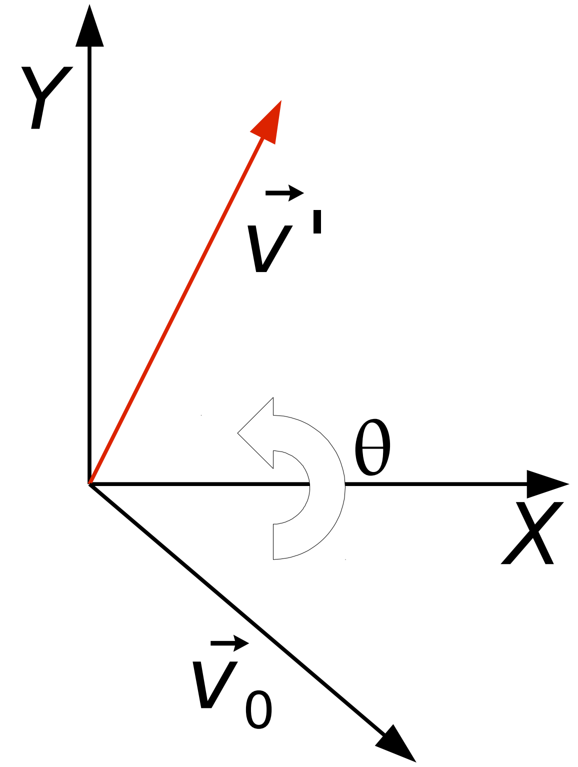


Geometric operations

- Simple matrix arithmetic
  - Rotations, reflections,...
  - Shearing, scaling,...

$$R_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$\vec{v}' = R_\theta \vec{v}_0$$



Numpy I

- Numpy adds fast array and matrix calculations to Python
  - http://numpy.scipy.org/

```
>>> from numpy import *

# Matrix initialization
>>> m=mat([[1,2], [3,4]])

# Matrix multiplication
>>> m*m
matrix([[ 7, 10],
        [15, 22]])

# Element-wise multiplication
>> m.A*m.A
array([[ 1,  4],
       [ 9, 16]]
```

```
>>> from numpy import *

# Array initialization
>>> m=array([[1,2], [3,4]])

# Matrix multiplication
>>> dot(m, m)
array([[ 7, 10],
       [15, 22]])

# Element-wise multiplication
>> m*m
array([[ 1,  4],
       [ 9, 16]])
```

Numpy II

- Indexing and slicing

```
>>> from numpy import *

>>> a=mat([[1,2], [3,4], [5,6]])    # Matrix initialization

>>> a[1]                            # Indexing (row)
matrix([[3, 4]])
>>> a[1,1]                          # Indexing (element)
4
>>> a[1:3, 1]                       # Slicing
matrix([[4],
        [6]])
>>> a[:,1]                          # Indexing (column)
matrix([[2],
        [4],
        [6]])
```

Numpy III

- Transpose, shape, svd

```
>>> from numpy import *
>>> b=mat([[1], [2], [3]])
>>> b.shape                          # Shape of the matrix
(3,1)
>>> b.T.shape                        # Transpose
(1,3)
>>> a.shape
(3,2)

>>> b.T*a                            # Matrix multiplication
matrix([[22, 28]])

>>> from numpy.linalg import *  # svd, det...
>>> v, s, wt=svd(a)              # Singular value decomposition
```

Numpy IV

- Determinant, universal functions

```
>>> a.T                        # Transpose
Matrix([[1, 3, 5],
        [2, 4, 6]])

>>> det(a[0:2, 0:2])           # Determinant
-2.0

>>> sqrt(1+a/2)                # Element-wise operations (sqrt, sin, cos, *,+...)
matrix([[ 1.        ,  1.41421356],
        [ 1.41421356,  1.73205081],
        [ 1.73205081,  2.        ]])

>>> diag([1,1,1])              # Diagonal matrix
array([[1, 0, 0],
       [0, 1, 0],
       [0, 0, 1]])
```

Numpy V

- Some more operations

```
>>> a
matrix([[1, 2],
        [3, 4],
        [5, 6]])

>>> sum(a)              # Total sum
21

>>> sum(a, 0)           # Sum along axis 0
matrix([[ 9, 12]])

>>> sum(a, 1)           # Sum along axis 1
matrix([[ 3],
        [ 7],
        [11]])
```

```
>>> a
matrix([[1, 2],
        [3, 4],
        [5, 6]])

>>> a.reshape((6,1)) # Reshaping
matrix([[1],
        [2],
        [3],
        [4],
        [5],
        [6]])
```